ATARI ST COMPUTER
Die Fachzeitschrift für ATARI ST- und TT-Anwender
Januar 91
DM 8,-
Ös. 64,-
Sfr. 8,-
1

MEGA STE
Erste Infos über Ataris neuen Rechner

NVDI
Neues VDI für ST/TT

Artworks
Calamus-Anwendungen

Orgatec
Was gab´s Neues in Köln?

SPS
Speicherprogrammierbare Steuerung

Neue Serie
Compiler-Bau

gw

Alle Jahre wieder

Auch auf die Gefahr hin, daß Sie jetzt entsetzt aufstöhnen, möchte ich in diesem Editorial auf fünf Jahre ST-Computer eingehen. Natürlich bietet sich zur Jahreswende immer dieses Thema an, aber ich denke, ein fünfjähriges Jubiläum sollte nicht ganz unter den Tisch fallen. Böse Stimmen behaupten ja, die ST-Computer könne jedes Jahr feiern, in dem sie sich gegen die Konkurrenz behaupten konnte. Da nützen aber auch keine Unkenrufe, denn es steht fest, daß die ST-Computer nach wie vor weltweit die größte Fachzeitschrift für die Atari ST/TT-Serie ist. Und das verdanken wir Ihnen, liebe Leser und Leserinnen. Die ST-Computer, eine Gemeinschaftsproduktion zweier Firmen [MAXON Computer (kompletter redaktioneller Inhalt) und Heim-Verlag (Werbung, Drucklegung etc.)], hat gegen Computer-Mediengiganten wie z.B. Markt&Technik auf dem ST-Markt durchaus bestehen können. Vielleicht auch gerade deswegen, weil nicht so ein riesiger Apparat dahintersteht und man somit etwas individueller arbeiten kann.

Zum Jubiläum haben wir auch noch zwei Leckerbissen für Sie, liebe Leser und Leserinnen. Zum einen wird der Preis der Monatsdisketten, die ja zweimonatlich erscheinen und alle Listings und Programme aus der ST-Computer enthalten, auf DM 12,- gesenkt, so daß man quasi für eine Ausgabe inkl. Listings und Programme DM 14,- bezahlt, zum anderen sind Kleinanzeigen ab der Ausgabe 2/90 kostenlos, sofern sie das vorgegebene Format einhalten und nicht gewerblichen Zwecken dienen. Näheres kann man in den News erfahren.

Ein frohes Fest und einen hoffentlich nicht eisglatten Rutsch ins neue Jahr wünscht Ihnen Ihre ST-Computer-Redaktion.

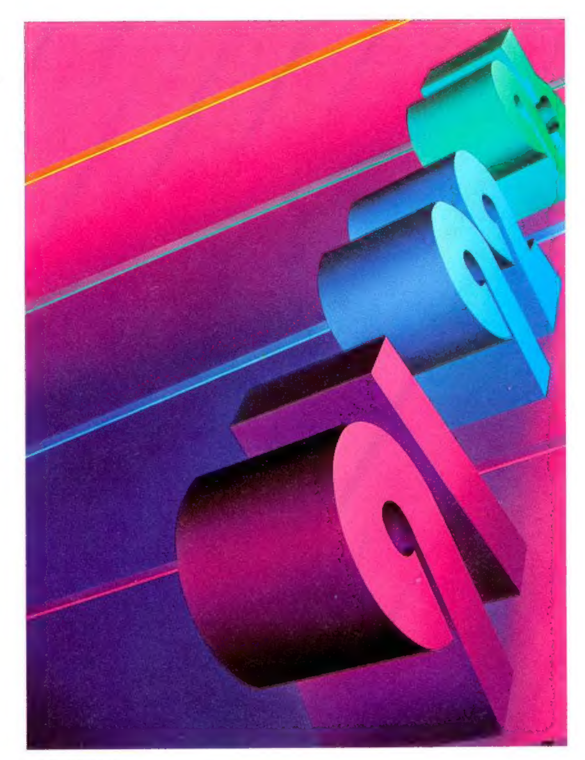
Harald Egel

I N H A L T

SOFTWARE

HARDWARE

PROGRAMMIERPRAXIS



NVDI

Neues VDI inklusive GDOS

Der „Software-Blitter" Turbo ST von Bela dürfte vielen ST-Besitzern bereits bekannt sein. Aus dem gleichen Hause ist nun das Programm NVDI erschienen, das ebenfalls dafür sorgen soll, der Bildschirmausgabe des ST Beine zu machen. Aber nicht nur das. NVDI hat eine ganze Menge mehr zu bieten. Neben einem höheren Kompatibilitätsgrad zu der immer größer werdenden Anzahl an ST-Programmen stellt NVDI ein völlig neu programmiertes GDOS inklusive neuem VDI dar.

Seite **18**

SPS

Seit ungefähr 20 Jahren, und in letzter Zeit immer stärker, geistert ein Begriff durch unsere Industrie. „Speicherprogrammierbare Steuerung" heißt dieses Zauberwort (im folgenden kurz „SPS" genannt), das viele Menschen in Entzücken versetzt, manchen Leuten aber auch etwas Unsicherheit einflößen kann. Dieser Artikel soll dazu dienen, Ihnen zu erklären, was SPS eigentlich ist, was man damit machen kann, worin die Vorteile liegen und schließlich: Wie kann ich mit meinem Atari-Computer die SPS-Programmierung kennenlernen? Der folgende Artikel und alle technischen Daten beziehen sich auf eine Siemens-STEP-5-SPS-Anlage, da diese doch sehr weit verbreitet und mittlerweile, besonders in der Ausbildung, zur Norm geworden sind.

Seite **51**



Compiler-Bau

Nachdem wir im Rahmen unseres C-Kurses bereits in der letzten Ausgabe auf das Thema Compiler-Bau eingegangen sind, möchten wir es mit unserer neuen Serie noch weiter vertiefen. Viele Programmierer sehen in einem Compiler immer noch ein fast mystisches Werkzeug, das die von ihnen geschriebenen Programme auf wundersame Weise von einer Hochsprache in Maschinencode übersetzt. Doch auch ein Compiler ist nur ein ganz normales Programm, dessen Funktionsweise jeder verstehen kann. Mehr darüber ab

Seite 134



Atari MEGA STE

Der neue ST

Atari sorgt dieses Jahr für eine Weihnachtsüberraschung in der ST-Gemeinde. Zwar schwirrten bereits seit der letzten CeBIT Gerüchte um einen Atari MEGA STE herum. Doch in den Chefetagen hüllte man sich in eisiges Schweigen. Genaues war nicht zu erfahren bzw. nicht für die Ohren der breiten Öffentlichkeit bestimmt. Jetzt wurde die neue Maschine auf der Comdex-Messe in den USA vorgestellt. Erste Informationen auf

Seite 23

GRUNDLAGEN

AKTUELLES

RUBRIKEN



In eigener Sache

Ab dieser Ausgabe gibt es einige Neuerungen bei der ST-Computer, die für alle Leser interessant sind. Zum einen sind ab sofort, quasi als Weihnachtsgeschenk für alle treuen Leser, alle privaten Kleinanzeigen kostenlos. Dadurch geben wir Ihnen die Möglichkeit, preiswert und effektiv Zubehör zum ST zu kaufen und verkaufen.
Die zweite Neuerung betrifft alle Programmierer und interessierten Leser, denn ab sofort kosten unsere Monatsdisketten nur noch DM 12,-. Auf einer Diskette sind alle Listings aus zwei Ausgaben gespeichert. Dadurch entfällt das lästige Abtippen. Der Clou bei der Sache: Versandkosten entfallen völlig. Dadurch kostet Sie ein Heft inklusive Monatsdiskette nur noch 14,- DM - ein guter Preis, wie wir finden. Die Monatsdisketten erscheinen alle zwei Monate, also jeweils im Februar, April, Juni, Juli (für die Doppelausgabe gibt es ausnahmsweise nur eine Diskette), Oktober und Dezember. Die Disketten können beim Heim-Verlag bestellt werden.

Hell-Satzbelichter

HELL bietet zwei speziell für das Atari-DTP-System angepaßte Hell-Imagesetter mit 396 mm (94 Pica) und mit 303 mm (72 Pica) Belichtungsbreite an. Die Anpassung an das Atari-System und die Entwicklung des entsprechenden Steuermoduls wurde von 3K Computerbild übernommen. Der Hell-Belichter ist somit auch in der Lage, Halbtonbilder in den präzisen Rastern von Retouche Professional in Höchstgeschwindigkeit auszugeben. Für eine DIN-A4-Seite in 1200 dpi beträgt die Ausgabezeit ca. 2,5 Minuten. Die Auflösung reicht bis 2400 dpi. Der Preis für den 72 Pica-Hell-Imagesetter einschließlich Software und Steuereinheit beträgt DM 56430,-. In die Steuereinheit ist eine 8-Bit-Grafikkarte integriert, die Software enthält Ausgabetreiber für Calamus, Retouche Professional und Didot Lineart.



3K Computerbild GmbH
Sassenfeld 71
W-4054 Nettetal 1
Tel. (02153) 60001

Neue MegaPaint II-Module

Ab sofort liefert der Berliner Software-Hersteller TommySoftware mehrere neue Module für den Einsatz mit MegaPaint II ab Version 3.0 aus:
Der neue TT-Rechner von Atari kann jetzt mit einem speziellen TT-Modul angesprochen werden. Die neue Rechnerarchitektur des TT erforderte eine spezielle Anpassung von MegaPaint. Das Modul ist für DM 399,- lieferbar. Weiterhin steht jetzt mit MegaConvert ein Dateikonverter zur Verfügung, der GEM-Metafiles in das MegaPaint-Vektorgrafikformat und umgekehrt konvertiert. Damit konnte ein häufig geäußerter Kundenwunsch erfüllt werden. MegaConvert kostet DM 149,-.
Mit einem speziellen ACC-Modul ist nun auch die Anwendung von Desktop-Accessories zusammen mit MegaPaint möglich. Das ACC-Modul kostet DM 99,-.
Alle Module können über den Fachhandel oder direkt über TommySoftware bezogen werden. Weitere Module sind derzeit in Planung.

TommySoftware
Selchower Straße 32
W-1000 Berlin 44
Tel. (030) 621-4063

Augur-Aktion und -Tool

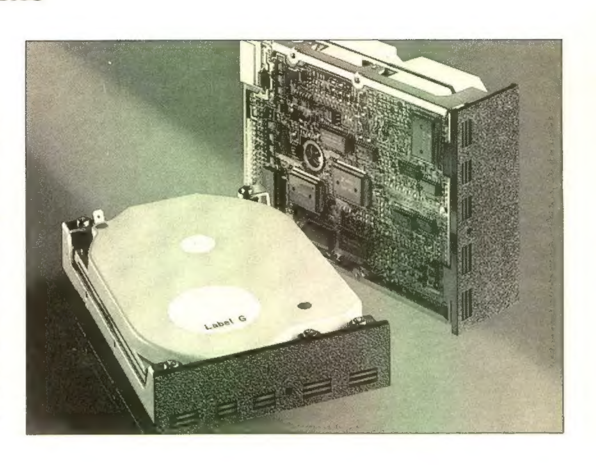
Das OCR-Programm Augur kann ein Jubiläum feiern. Seit der Einführung von Augur im Jahr 1988 wurden bis heute 1000 Exemplare verkauft. Wenn man den Preis von beinahe Fr 3000,- berücksichtigt, ist dies ein beachtlicher Erfolg. Die Marvin AG feiert dieses Jubiläum und bietet ab sofort einen Jubiläumspreis von DM 1481,- / sFr. 1111,- statt sFr. 2990,- an, limitiert bis zum 28.2.1991. Anwendern, die Syntex profesionell nutzen, bietet Marvin die Möglichkeit, auf Augur umzusteigen. Für sFr. 950,- / DM 1266,- ist für registrierte Syntex-Benutzer das Flaggschiff unter den Texterkennungsprogrammen, Augur 1.5, erhältlich. Auch diese Aktion ist bis zum 28.2.1991 begrenzt.
Das unter Augur-Anwendern beliebte Utility, Augur-Tool, kann neu auch mit Syntex verwendet werden. Dieses Hilfsprogramm erlaubt es, bestehende OCR-Bibliotheken von Augur oder Syntex zu verwalten, zu kombinieren und zu bearbeiten, wobei die Verwaltung weit über die Funktionen anderer OCR-Programme hinausgeht. Augur Tool kostet sFr. 90,- / DM 120,-.

Marvin AG
Fries Straße 23
CH-8050 Zürich
Tel. (01) 3022179

Telebox wird international

Die Deutsche Bundespost Telekom wird in Kürze ihr Mitteilungssystem Telebox mit dem Mailbox-Systemen der USA (AT&T) ATTMail und der Schweiz (PTT) arCom verknüpfen. Die entsprechenden Vereinbarungen wurden bereits getroffen, weitere Vereinbarungen mit Dienstleistungsanbietern anderer Länder stehen unmittelbar vor dem Abschluß. Danach wird ein dichtes Netz an Verkehrsbeziehungen vorhanden sein, das einen kompatiblen Informationsaustausch von Benutzern in öffentlichen und privaten Versorgungsbereichen erlauben wird. Grundlage für die offene Kommunikation ist die Standardisierung im Bereich von OSI (Open Systems Interconnection). Zu diesen Standards gehören auch die der Serie X.400 des CCITT (Comite Consultatif International Telephonique et Telegraphique) für Mitteilungsübermittlung. Sie gestatten Benutzern von DV-Systemen und Computern, Informationen mit beliebigen Partnern in der Welt auszutauschen. Die Telekom wird die Dienstleistung Telebox mit dem System Telebox-400-MT in diese Richtung weiterentwikkeln.

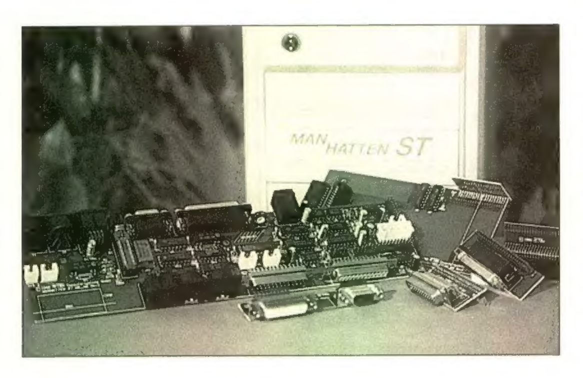
Info:
Deutsche Bundespost Telekom
Postfach 5000
W-6100 Darmstadt
Tel. (06151) 83-2005



Zugriffszeit verringert

Die Modelle der Fujitsu M2610-Serie mit der Bezeichnung ESA, einer Laufwerk-Palette von 90 MByte bis 180 MByte Speicherkapazität, verfügen über eine SCSI-Schnittstelle. Die im Laufwerk integrierte Controler-Intelligenz und ein 34 kByte großer Datenpuffer erlauben Transferraten von 2,5 MB synchron und 1,5 MB asynchron. Der Entwicklung Rechnung getragen wurde auch bei den „ET"-Modellen der M-2610-Serie mit AT-Bus-Interface (IDE-Interface). Die in die Laufwerke integrierten 56 kByte großen Datenpuffer gestatten Transferraten bis zu 7,4 MByte vom Puffer zum Host. Die zusätzlichen Betriebsarten Idle und PowerSave unterstützen ideal den Betrieb in Laptop-Computern. Modelle der M2610-Serie mit „ESA-M"-Bezeichnungen sind auch für Macintosh und Atari ST erhältlich. Alle Laufwerke dieser Serie erlauben sowohl horizontalen als auch vertikalen Einbau und bieten trotz kleiner Abmessungen überragende Leistung.

Info:
Fujitsu Deutschland GmbH
Frankfurter Ring 211
W-8000 München 40
Tel. (089) 32378-0



Manhatten ST

Das neue Modell des Manhatten ST zeigt besonders im Schnittstellenbereich entscheidende Änderungen. Eine hardwaremäßige Bildschirmschonschaltung verhindert das Einbrennen des Monitorbildes. Die neuen leistungsstarken Adapterplatinen führen alle Schnittstellen ohne „Kabelsalat" zu einer speziell beschrifteten Schnittstellenabdeckung nach außen. Zusätzlich zu den bisher bereits vorhandenen Schnittstellen sind nun auch standardmäßig DMA und SCSI herausgeführt. Der Sub-D-Anschluß für Farb- und s/w-Monitore ist an der Schnittstellenabdeckung schaltbar, Tastatur und Maus sind durch ein Schloß im Tower-Gehäuse abschließbar und so vor Fremdbedienung geschützt. Der ROM-Port kann intern und extern bestückt und dann softwaremäßig umgeschaltet werden. Außerdem sind die Schnittstellen MIDI in und out vorhanden. Der Manhatten ist für den Anschluß eines Scanners vorbereitet (I/O Pin 17 beschaltet), besitzt interne Stromversorgungsanschlüsse und 8 softwaremäßig ansprechbare Schaltausgänge. Damit ist der Manhatten ST unübertroffen „kontaktfreudig". All diese vorher mit Aufpreisen versehenen Optionen sind nun im Preis des neuen Manhatten von DM 5980,- enthalten.

Tetra Computersysteme GmbH
Neuer Markt 27
W-5309 Meckenheim
Tel. (02225) 17081

Calligrapher

Die Hamburger Firma SciLab hat die Exklusivdistribution des Programms „Calligrapher" für Deutschland und Österreich übernommen. Wer sich für das Programm interessiert, sollte sich also ab sofort direkt an SciLab wenden.

SciLab GmbH
Isestraße 57
W-2000 Hamburg 13
Tel. (040) 4603702

Templemon V1.2

Der Templemon ist in der Version 1.2 jetzt auch auf dem TT lauffähig. Als weitere Neuerung wurde ein History-Puffer eingebaut. Die Anleitung zum Programm kostet DM 20,-. Die Bezugsadresse lautet

Johannes Hill
Alicenstraße 30
W-6100 Darmstadt

Quantos

Speziell für ST-Besitzer, die nicht bereit sind, mehrere 100 kByte des Hauptspeichers an Hintergrundprogramme abzutreten, wurde Quantos entwickelt. Durch die Programmierung in schnellem Assembler begnügt sich diese Betriebssystemerweiterung mit weniger als 20 kByte des freien RAM. Weniger bescheiden gibt sich Quantos jedoch in seiner Funktionsvielfalt. Neben den liebgewordenen Features wie Mausbeschleuniger, Bildschirmschoner und Uhrzeitanzeige finden sich auch eine flexible, resetfeste RAM-Disk sowie eine Hardcopy für 24-Nadel- und Laserdrucker, die es in sich hat: nach dem Druck auf Alternate/Help setzt man bequem mit der Maus einen Bildschirmausschnitt, der dann in wählbarer Größe und Lage zu Papier gebracht wird. Damit ist es endlich möglich, nur das Inhaltsverzeichnis statt des gesamten Desktops zu drucken. Ein Drucker-Spooler, der auch mit SIGNUM! zusammenarbeitet, verhindert lange Wartezeiten ebenso wie das abschaltbare Diskverify. Ein Wecker und weitere Hilfen vervollständigen den Funktionsumfang von Quantos, das mit sämtlichen Funktionen resetfest ist. Der Preis für diesen Winzling beträgt DM 49,-.

Bernd Blank
Friedrichstaler Allee 6
W-7513 Stutensee-Fr.
Tel. (07249) 4422

Letzte Meldung: Atari senkt die TT-Preise

Pünktlich zum Weihnachtsfest hat Atari die Preise der TT-Serie gesenkt. Man kann somit relativ kostengünstig zu einem 68030-Rechner kommen. Es dürfte momentan sogar der billigste Computer mit diesem Prozessor sein. Auch ist man nicht mehr gezwungen, einen TT mit Monitor zu kaufen. Allerdings bietet Atari selbst einen 14"-Farb- für DM 998,- und einen 19"-Monochrommonitor für DM 2198,- an. Die Preise für einen TT inkl. 48MB-Festplatte und ohne Monitor:

TT mit 4MB RAM	DM 4298,-
TT mit 6MB RAM	DM 4798,-
TT mit 8MB RAM	DM 5298,-

Alle Atari-Fachhändler

BTX-Börsenmanager 4.0

Die neueste Version des BTX-Börsenmanagers wurde um einige Neuerungen erweitert. Es können bis zu 980 Aktien mit wöchentlichen Kursen 4 Jahre rückwirkend im Speicher gehalten werden. Die Indexbildung wurde wesentlich beschleunigt. Zu den Gesamtmarkt-, Blue-Chip- und Branchenindexen, die bisher gebildet wurden, kann man bei Auslandswerten Länderindexe bilden. Die Zeitraumdefinition im Analyseteil, um eigene Listen auszugeben, wurde wesentlich verbessert. Die Kurseingabe per Hand ist vollkommen neu überarbeitet, wobei man einen BTX-Textmonitor zur Durchsicht nach neuen Aktien bzw. nicht gefundenen Kursen integriert hat. Kursveränderungen zur Vorwoche über 10% werden zur Kontrolle angezeigt. Der Preis des Programms soll ab 1.1.91 in der Version 4.0 DM 189,- betragen. In diesem Preis sind zirka 1500 Datensätze mit deutschen, in Deutschland gehandelten Auslandswerten und den amerikanischen Standard & Poors-500-Werten mit der entsprechenden Auslesedatei für den Drews-BTX-Manager enthalten. Eine Demo-Version ist für DM 10,- erhältlich.

Thomas Bopp Softwarevertrieb
Wintersteinstraße 21
W-6361 Niddatal 1
Tel. (06034) 7961

Practical Solutions

Die Firma Omikron hat den Vertrieb für die Produkte der amerikanischen „Practical Solutions" übernommen - intelligente Hardware-Lösungen für die kleinen Probleme des Alltags. Mit dem „Mouse Master" beispielsweise kann man Joystick und Maus zusammen an einen Port anschließen; statt hin- und herzustöpseln, genügt dann einfach ein Schalterklick. Mouse Master kostet DM 79,-. Der „Drive Master" schaltet mehrere Diskettenlaufwerke um. So wird ein drittes Laufwerk am ST möglich. Das ist besonders nützlich, wenn man mit einem MS-DOS-Emulator arbeitet und ein 54"-Laufwerk zusätzlich anschließen will. Drive Master kostet DM 89,-. Ein MultiSync-Monitor ist eine tolle Sache - aber der Ton fehlt! Das „Tweety-Board" schafft für DM 99,- Abhilfe. Die schnurlose Maus von Practical Solutions schafft Ordnung und liegt besonders gut in der Hand. Im Preis inbegriffen ist ein universell einstellbarer Maustreiber, mit dem nicht nur „Quick-Mouse", sondern auch „Slow-Mouse"-Einstellungen (für genaue Positionierung) möglich sind. Die Maus kostet DM 198,-.

Omikron.Software
Sponheimstraße 12
W-7530 Pforzheim
Tel. (07231) 356033

Einigungsvertrag auf Disk

Den Einigungsvertrag über die Herstellung der Einheit Deutschlands mit sämtlichen Anlagen gibt es jetzt für den Atari ST als Diskettenversion: Für den schnellen und universellen Zugriff auf den im Volltext erfaßten Vertrag mit Volltext-Retrieval und Hypertextstrukturen sorgt das bekannte 1st_Card von LogiLex. Die Lieferung enthält auf 4 Disketten knapp 3 MByte Daten, die auf Harddisk zu installieren sind. Begleitend gibt es den Vertragstext mit sämtlichen Anlagen und ausführlichem Gesetz- und Sachregister als knapp 700 Seiten starkes Buch der 'jur-pc-Schriftenreihe' aus dem Verlag MediConsult. Alles zusammen kostet nur DM 98,- und ist bei LogiLex ab sofort erhältlich. Bei Vorkasse (V-Scheck) erfolgt die Lieferung frei Haus, bei Nachnahmebezug zzgl. DM 6,- Versandkosten.

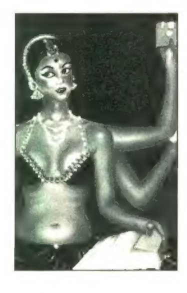
LogiLex - Gerhard Oppenhorst
Eifelstraße 32
W-5300 Bonn 1
Tel. (0228) 658346

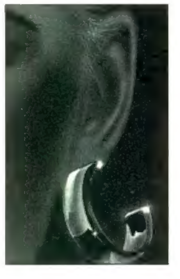




AKTUELLES

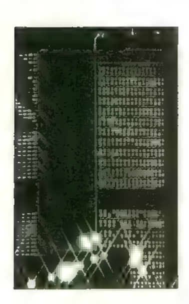
SOFTWARE





DFÜ

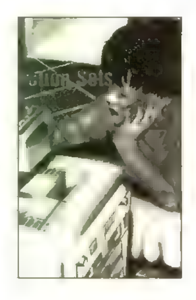




PROGRAMMIERPRAXIS



HARDWARE-PROJEKTE

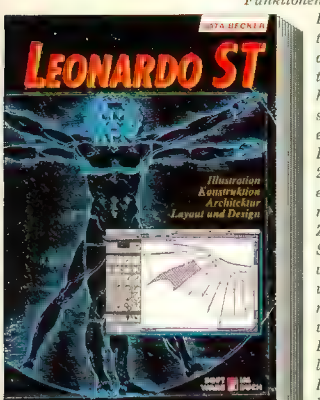


HARDWARETESTS

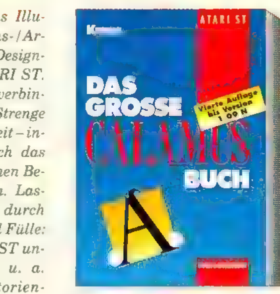
FÜR EINSTEIGER UND AUFSTEIGER!

DAS EINFACH GENIALE PROGRAMM: LEONARDO ST!

LEONARDO ST ist das Illustrations-/Konstruktions-/Architektur-/Layout- und Design-Programm für den ATARI ST. Durch LEONARDO ST verbinden Sie mathematische Strenge mit künstlerischer Freiheit – inklusive der Freiheit, sich das Programm auf Ihre eigenen Bedürfnisse zuzuschneiden. Lassen Sie sich beflügeln durch Funktionen in Hülle und Fülle: LEONARDO ST unterstutzt Sie u. a. durch objektorientiertes Zeichnen mit hoher Präzision: verschiedene Kurvenelemente für den Freihand-Entwurf; 250 Zeichenebenen: eine Undo-Funktion, mit der Sie alle (!) Zeichenvorgänge Schritt für Schritt widerrufen können, umfangreiche Bemaßungsfunktionen und Werkzeuge zum Konstruieren. Natürlich können Sie auch Flächen füllen, Symbole erstellen und einfügen sowie Vektor-Schriften für Illustrationen nutzen. Das Programm unterstützt gängige 9- und 24-Nadeldrucker, Laserdrucker, PostScript-Drucker, HP-kompatible Plotter und Großbildschirme.



LEONARDO ST
DM 99,-
ISBN 3-89011-818-6



Bestes Computerwissen allein reicht noch nicht aus für perfekte DTP-Ergebnisse. Auf der einen Seite sollte man alle Funktionen und Möglichkeiten des entsprechenden Programms kennen und beherrschen, aber auch in Sachen Typografie und Gestaltung auf der Höhe sein. Was Calamus angeht, gibt es ein Buch, das Ihnen all dieses Wissen vermittelt: Das große Buch zu Calamus. Lesen Sie, wie Sie mit Calamus gekonnt Ihre eigenen Drucksachen erstellen - vom Starten des Programms bis zum Drucken. Dieses Buch sagt Ihnen alles über die Benutzerführung, verschiedene Layouts, individuelle Suchpfade, Druckertreiber sowie Text- und Grafikrahmen.

Das große Calamus -Buch
392 Seiten, DM 39,-
ISBN 3.89011-346-X

SCHLÜSSEL ZU STARKEN SPIELEN

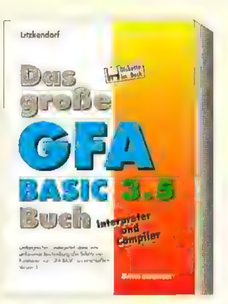
Falls Sie Adventures lieben, haben wir genau das richtige für Sie: Unsere aktuellen Bände zu „Space Quest". zu „Leisure Suit Larry" und zu „King's Quest". Folgen Sie mit uns den Helden Roger Wilco, Larry Laffer und Graham auf ihren erlebnisreichen Pfaden durch die Welt der Abenteuer. Retten Sie Ihre Freunde aus ausweglosen Situationen, treffen Sie in Schlüsselszenen die richtigen Entscheidungen, und sammeln Sie unterwegs die richtigen Gegenstände. Alle drei Bücher halten dazu praktische Komplettlösungen bereit, ohne Ihren Entdeckergeist und damit den Spaß am Spielen einzuschränken.



Darr
Die Space Quest Story
148 Seiten
DM 19,80
ISBN 3-89011-280-3

Müller/Schuchardt
Die Larry Story
160 Seiten, DM 19,80
ISBN 3-89011-188-2

Koj
Die King's Quest Saga
152 Seiten, DM 19,80
ISBN 3-89011-291-9



In diesem Buch finden Sie alles, was Sie für die fortgeschrittene Programmierung benötigen: die praktische Anwendung der einzelnen GFA-Befehle der aktuellen Version 3.5. das Einbinden von Systemroutinen, alles Wissenswerte zur Programmstruktur sowie exemplarische Grafik- und GEM-Programmierung – alles immer anhand praktischer Beispiele erklärt. Selbstverständlich enthält das Buch auch eine komplette Beschreibung aller Optionen des 3.5-Compilers und eine Ubersicht sämtlicher Fehlermeldungen. Das große GFA-Buch – ein Nachschlagewerk mit dem Know-how eines Profis.

Litzkendorf
Das große GFA-BASIC-Buch
Hardcover, inklusive Diskette
899 Seiten, DM 59,-
ISBN 3-89011-363-X



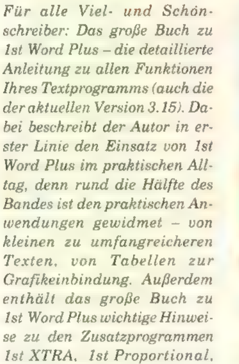
Mit dem großen ST-Handbuch lösen Sie alle Probleme rund um den ST sozusagen im Handumdrehen. Bei Problemen mit dem Desktop genauso wie bei der Tastatur oder den Schnittstellen. Dabei werden Sie natürlich noch jede Menge interessanter Neuigkeiten über Ihren Rechner erfahren und auf eine Reihe nützlicher Tricks stoßen. Zahlreiche Tips zur Pflege Ihres Rechners und zum „Rechner-Tuning" gibt Ihnen dieses Buch genauso weiter wie viele kleine Reparatur-Hinweise. Ein Nachschlagewerk, das Sie immer wieder einmal benutzen werden - nicht zuletzt dank seiner klaren Gliederung. Der ST/E wird natürlich ebenfalls behandelt.

Liesert
Das große ST-Handbuch
Hardcover, 377 S., DM 49,-
ISBN 3-89011-273-0

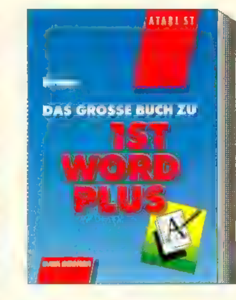


Ob Diskettenoperationen, Texteinstellungen, Fußnotenverwaltung, Spaltensatz, Druckausgabe oder Bildoperationen mit diesem Buch lernen Sie Signum/Signum II von seiner besten Seite kennen. Auf das umfassende Thema Druckerausgabe geht der Autor dabei besonders ausführlich ein: Vom Druckfonteditor bis zum perfekten Ausdruck sagt er Ihnen, worauf Sie achten müssen. So werden hier alle Features von Signum! — aber auch von Scarabus 2.0 und Headline 3.0 — anhand zahlreicher praktischer Ubungen beispielhaft erklärt. Das große Signum!-Buch: praxisorientierter Einstieg und umfassendes Nachschlagewerk.

Kraus
Das große Signum!-Buch
346 Seiten, DM 39,-
ISBN 3-89011-313-3



Für alle Viel- und Schönschreiber: Das große Buch zu 1st Word Plus – die detaillierte Anleitung zu allen Funktionen Ihres Textprogramms (auch die der aktuellen Version 3.15). Dabei beschreibt der Autor in erster Linie den Einsatz von 1st Word Plus im praktischen Alltag, denn rund die Hälfte des Bandes ist den praktischen Anwendungen gewidmet – von kleinen zu umfangreicheren Texten, von Tabellen zur Grafikeinbindung. Außerdem enthält das große Buch zu 1st Word Plus wichtige Hinweise zu den Zusatzprogrammen 1st XTRA, 1st Proportional, 1st Mail, zum Profi-Text-Modul sowie viele Tips & Tricks.



Kraus
Das große Buch zu 1st Word Plus
314 Seiten, DM 39,-
ISBN 3-89011-347-8

SOFORT BESTELLEN...

...bei DATA BECKER, Merowingerstraße 30, 4000 Düsseldorf 1

Liefern Sie mir bequem nach Hause:

Ich zahle (zzgl. DM 5,- Versandkosten, unabhängig von der bestellten Stückzahl)
- ☐ **per Nachnahme**
- ☐ **mit beiliegendem Verrechnungsscheck**

Name

Straße

PLZ/Ort

DATA BECKER

Merowingerstr. 30 • 4000 Düsseldorf 1 • Tel. (0211) 310010





RELAX



GRUNDLAGEN



INHALT







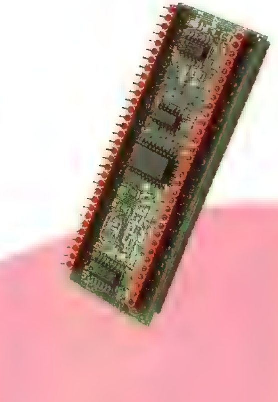
QUICKTIPS

ST-REPORT

BÜCHER

Eine kleine Sensation läßt TOS-Her(t)zen schneller schlagen. Der AdSpeed ST.





Der AdSpeed ST, ein neues Beschleuniger-Board, macht dem ATARI ST jetzt mal richtig Beine. Beim Arbeiten mit CAD und DTP zum Beispiel. Mit HighTech vom Kleinsten - 16 MHz auf nur 86 x 29 mm. Und 32 kByte RAM (exakt: 16 kByte data/ instruction cache, 16 kByte tag cache memory). Dabei reicht ein kleiner Eingriff und der AdSpeed ST ist installiert. Jene freundlichen Fachhändler dort links übrigens, die helfen Ihnen gern. Fürs Umschalten der Taktrate zwischen 8 und 16 MHz sorgt ein eigenes Accessory. Oder ein Hotkey. Oder ein Schalter. Bei gleichzeitiger Aktivierung/ Deaktivierung des Cache und dies sogar im laufenden Programm. Das garantiert 100%ige Kompatibilität mit allen Programmen. Eine Sache, die nur der AdSpeed ST drauf hat. Außerdem bei unserem Beschleuniger-Board exklusiv inklusive: die 68000er CMOS-CPU. Eine Fast-ROM-Option. Eine einblendbare Geschwindigkeitsanzeige. Und als Tüpfelchen auf dem i: Quick ST, zur Beschleunigung der Bildschirmausgabe.
Mehr Produktinfo? Senden wir Ihnen gern zu - Karte oder Anruf genügt. Have a nice day.

PS: Händleranfragen erwünscht.

Ihre AdSpeed-Händler:

DATAPLAY
Bundesallee 25
1000 Berlin 31

WBW Service
Sielwall 87
2800 Bremen 1

Chemo-Soft
Nadorster Straße 81
2900 Oldenburg

COM DATA
Am Schiffgraben 19
3000 Hannover 1

GengTec Gerald Geng
Teichstraße 20
4020 Mettmann

3K Computerbild GmbH
Sassenfeld 71
4054 Nettetal 1

CBS GmbH
Tecklenburger Straße 27
4430 Steinfurt

Hard & Soft A. Herberg
Obere Münsterstraße 33-35
4620 Castrop-Rauxel

CSA ComputerSysteme
Hüttenstraße 56
4650 Gelsenkirchen

Rolf Rocke Computer
Auestraße 1
5090 Leverkusen 3

Catch Computer
Ludwigsallee 1b
5100 Aachen

Computer Behnck GbR
Von-den-Driesch-Str. 89
5300 Bonn 1

Eickmann Computer
In der Römerstadt 249
6000 Frankfurt 90

WAVE Computersysteme
Südanlage 20
6300 Gießen

FSE Frank Strauß Elektronik
Schmiedstraße 11
6750 Kaiserslautern

Comp & Phone
Alleenstraße 66
7312 Kirchheim/Teck

Wacker Systemelektronik
Bachstraße 39
7500 Karlsruhe 21

Resin - Büro mit System
Am Dreispitz 6
7852 Binzen

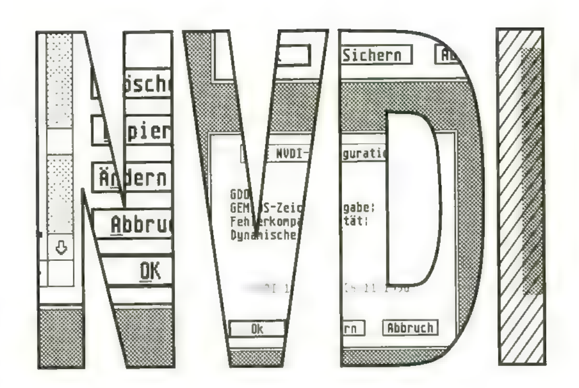
Uhlenhuth GmbH
Albrecht-Dürer-Platz 2
8720 Schweinfurt

MMAD



ICD Europe GmbH, PF 13 17, D- 6056 Heusenstamm, Telefon (0 61 04) 64 03, Fax 67 58 1

Ein neues GDOS braucht das Land



Der „Software-Blitter" Turbo ST von Bela dürfte vielen ST-Besitzern bereits bekannt sein. Aus dem gleichen Hause ist nun das Programm NVDI erschienen, das ebenfalls dafür sorgen soll, der Bildschirmausgabe des ST Beine zu machen. Aber nicht nur das. NVDI hat eine ganze Menge mehr zu bieten.

Neben einem höheren Kompatibilitätsgrad zu der immer größer werdenden Anzahl an ST-Programmen stellt NVDI ein völlig neu programmiertes GDOS inklusive neuem VDI dar. Es ist zudem noch kompatibel zu GEM 2.x, das ja leider nur über eine holländische Firma für den ST erhältlich ist. Aber vielleicht sollte man zunächst einmal erklären, was es mit Begriffen wie VDI, AES und GDOS eigentlich auf sich hat.

AES, VDI, GDOS, sonst noch was?

Die grafische Oberfläche GEM des ST setzt sich aus mehreren Bestandteilen zusammen. Da ist zunächst einmal das AES (Application Environment Services). Dieser Teil des GEM ist für die Verwaltung komplexer Funktionen zuständig. Hierzu zählen unter anderem die Bedienung der Menüleiste und der Fenster sowie das Zeichnen und Verwalten von Dialogboxen.

Für seine Arbeit benötigt das AES gewisse grafische Grundfunktionen, die es beispielsweise ermöglichen, Rechtecke zu zeichen (Umrahmung von Dialogboxen) oder Bildschirmbereiche zu invertieren (Selektieren von Knöpfen). Diese Routinen werden vom VDI (Virtual Device Interface) zur Verfügung gestellt. Das VDI stellt somit die unterste Ebene der Bildschirmroutinen dar. (Zwar greift das VDI seinerseits für einfache grafische Funktionen auf die sogenannten LINE-A-Routinen zurück, jedoch soll hierauf nicht näher eingegangen werden, da diese Routinen aus Kompatibilitätsgründen nicht mehr verwendet werden sollten.)

Theorie und Praxis

Eine der großen Ideen hinter GEM war (oder ist vielleicht immer noch), eine bedienerfreundliche Oberfläche zur Verfügung zu stellen, unabhängig von Ein- oder Ausgabegeräten. Man kann also seine Eingaben über beliebige Geräte vornehmen und das Ergebnis z.B. auf dem Bildschirm oder auf einem Drucker ausgeben lassen. So verhält es sich zumindest in der Theorie. In der Praxis sieht es beim Atari ST leider ein wenig anders aus. Da es Atari aus Platzgründen nicht möglich erschien, das komplette VDI mit in den 192kB ROM des ST unterzubringen, wurden einige VDI-Funktionen weggelassen. Hierbei handelt es sich in erster Linie um Routinen, die für die Verwaltung neuer Zeichensätze benötigt werden. Außerdem beinhaltet das VDI des ST nur einen Treiber für den Bildschirm und besitzt keine Routinen zur Verwaltung weiterer Gerätetreiber, so daß das GEM um eine wichtige Eigenschaft beraubt wurde.

Um nicht ganz auf das vollständige VDI verzichten zu müssen, wird von Atari ein Programm namens GDOS (Graphics Device Operating System) zur Verfügung gestellt, das das VDI des ST um die fehlenden Funktionen ergänzt und bei Bedarf nachgeladen werden kann.

Weiter im Text (weiter im Test)

Leider hat sich gezeigt, daß bei installiertem GDOS die Geschwindigkeit der GEM-Bildschirmausgabe geringer wird. (Die abgedruckte Tabelle mit Geschwindigkeitsvergleichen zeigt dies am Beispiel des Atari GDOS V1.1. Ein Kommentar erübrigt sich.) Dies dürfte der Hauptgrund sein, warum das GDOS nicht gerade zu den beliebtesten residenten Programmen für den ST zählt.

Mit NVDI steht nun ein GDOS zur Verfügung, das insbesondere die Geschwindigkeitsnachteile der bisherigen GDOS-Versionen wettmachen will. Es enthält nicht nur die fehlenden VDI-Routinen (also das eigentliche GDOS), sondern stellt vielmehr ein komplett neu programmiertes VDI dar. Und in der Tat kann sich die Geschwindigkeit von NVDI sehen lassen. Ein Urteil können Sie sich selbst bilden, wenn Sie die Ergebnisse der Geschwindigkeitstests vergleichen.

Geschwindigkeit ist keine Hexerei

Um beurteilen zu können, welchen Einfluß ein Programm auf die Geschwindigkeit der Bildschirmausgabe hat, bieten sich Messungen mit dem PD-Programm *Quickindex* (neuere Version nicht mehr PD!) an. Dieses Programm erlaubt den

Vergleich verschiedener Systemkonfigurationen. Die Tabelle zeigt Vergleichsdaten von Tests, die auf einem Atari ST mit Monochrommonitor und TOS 1.4 sowie auf einem TT in der hohen ST-Auflösung durchgeführt wurden. Um die Geschwindigkeitssteigerung durch NVDI besser einordnen zu können, wurde Turbo ST 1.8 als Referenz herangezogen. Weiter wurde zum Vergleich noch mit dem NVDI-Testprogramm getestet (Näheres darüber finden Sie im Kasten).

Es ist leicht zu erkennen, daß NVDI nicht nur relativ zur „normalen" Geschwindigkeit unter TOS 1.4 gut abschneidet (dies ist auch keine Kunst), sondern auch im Vergleich mit Turbo ST in manchen Punkten sogar schneller ist.

"Kaotische" Zustände

Da das gesamte GEM-System (also AES, VDI und GDOS) interaktiv funktioniert, ist natürlich ein Teil nur so gut wie die anderen. Deshalb wollen wir Ihnen auch nicht verheimlichen, daß auch in Richtung AES seit langem Optimierungen stattgefunden haben. Leider sind diese Änderungen meist nur auf Insider-Kreise beschränkt, da sie in keiner Form kommerziell angeboten werden. Vielleicht passiert das ja noch irgendwann einmal, wenn auch Atari momentan kein Interesse daran zeigt, Verbesserungen inoffizieller TOS-Versionen in die eigenen Entwicklungen zu übernehmen. Paradebeispiel für eine solche verbesserte TOS-Version stellt das sogenannte KAOS-TOS von Andreas Kromke, Dirk Katzschke und Normen Kowalewski dar. Dieses wurde in Form verschiedener TOS-Patches in der Zeitschrift *c't* vorgestellt.

NVDI profitiert besonders von der Optimierung der AES-Routinen, die im KAOS-TOS vorgenommen wurden. Eine Zusammenarbeit von KAOS-TOS und NVDI sorgt deshalb für einen zusätzlichen Geschwindigkeitsgewinn. Wünschenswert wäre es, wenn beide ins Betriebssystem integriert würden und als Ersatz für das von Atari gelieferte AES-VDI benutzt werden könnten. Aber das ist zur Zeit wirklich nur ein Wunschtraum, da sich Atari nicht zur Akzeptanz verbesserter TOS-Versionen durchringen kann.

Schnell, schneller, am schnellsten

Wie Sie den Tabellen entnehmen können, wird durch den Einsatz von NVDI eine deutliche Geschwindigkeitssteigerung erreicht.

Wünscht man sich einen schnelleren ST auf dem Tisch, baut sich der finanzkräftigere Anwender eine sogenannte Accelerator-Karte in seinen ST ein. Davon gibt es auf dem Markt inzwischen eine ganze Reihe in den unterschiedlichsten Preislagen. Die Palette reicht vom schneller getakteten 68000-Prozessor bis zum Einbau eines „großen Bruders" wie dem 68020 oder sogar dem 68030, wie er im TT zu finden ist. Übrigens haben wir NVDI ebenfalls mit einem 68010-Prozessor, dem MAXON Board20 (68020) und der pro-VME 68030-Karte getestet. NVDI lief ohne Probleme.



Mittels Accessory läßt sich NVDI konfigurieren.

Geht es um eine Beschleunigung der Grafikroutinen, zieht ein normaler ST mittels NVDI in vielen Punkten mit dem TT gleich und übertrifft dessen Geschwindigkeit sogar. Und dabei (man höre und staune) liegt der Preis von NVDI deutlich unter dem eines TT.

Auch auf einem TT zeigt NVDI seine Wirkung. Es läuft auf allen TT-Auflösungen. Die Vergleichstabelle dokumentiert den Geschwindigkeitszuwachs durch NVDI zwar mit Zahlen, aber man muß es einfach selber gesehen haben, wie blitzartig Dialogboxen nun auf dem Bildschirm erscheinen.

NVDI + Turbo ST = BOMBEN

Nun könnte man auf die Idee kommen, sich folgende Systemkonfiguration zusammenzustellen: KAOS-TOS, NVDI und zusätzlich Turbo ST. In dieser Kombination könnte nochmals eine Geschwindigkeitssteigerung bei der Text- und Grafikausgabe erreichbar sein.

Leider ist es nicht möglich, mit der angegebenen Programmkombination zu arbeiten. NVDI und Turbo ST vertragen sich nämlich nicht miteinander. Möglicherweise hält sich Turbo ST nicht an die VDI-Konventionen. Da sowohl NVDI als auch Turbo ST von Bela vertrieben werden, kann man nur hoffen, daß hier das letzte Wort noch nicht gesprochen ist.

Kompatibilität

Werden Teile des Betriebssystems neu programmiert, sind Kompatibilitätsprobleme nicht auszuschließen. Gerade das VDI enthält viele elementare Routinen, die von eigentlich jedem grafisch orientierten Programm genutzt werden. Die Neuprogrammierung aller VDI-Funktionen ist somit ein Prüfstein für Programme, die das VDI nutzen. Programme, die sich beim Aufruf des VDI nicht an die vorgegebenen Richtlinien halten, können Schiffbruch erleiden. Hier heißt es insbesondere für den Assembler-Programmierer, wirklich alle Register, die durch einen VDI-Aufruf verändert werden könnten, vorher zu sichern. Turbo ST-Besitzer können hiervon ein Lied singen. Der deutliche Geschwindigkeitszuwachs wurde bei diesem Programm dadurch erreicht, daß ein großer Teil der für die Ausgabe auf dem Bildschirm verantwortlichen Betriebssystemfunktionen einfach ersetzt wurde. Daß dieses Vorgehen nicht unbedingt sauber war, bescheinigen einige (eventuell ebenfalls unsauber programmierte) Programme, die in Verbindung mit Turbo ST nicht mehr einwandfrei funktionieren.

Die Suche nach Programmen, die sich nicht mit NVDI vertragen, war, von Turbo ST einmal abgesehen, erfreulicherweise nicht von Erfolg gekrönt. Wie sauber es programmiert ist, bewies sogar ein Testlauf mit dem Atari ST-Emulator *Chamäleon* auf einem Amiga.

Konfiguration

Auf der mitgelieferten Diskette befindet sich ein Programm, das die gesamte Installation von NVDI übernimmt. Also keine Probleme für Anfänger und reine Anwender.

In manchen Fällen dürfte es nützlich sein, daß Teile von NVDI über ein mitgeliefertes Accessory abgeschaltet werden können, um Kompatibilitätsprobleme zu mindern. Wichtig ist in diesem Zusammenhang vor allen Dingen die Möglichkeit, das GDOS abzuschalten, da manche Programme sich hiermit nicht vertragen. Komplett abschalten läßt sich NVDI übrigens nicht. Stößt man also auf ein Pro-

gramm, das sich mit NVDI gar nicht verträgt, bleibt nichts anderes übrig, als neu zu booten.

Neben der eigentlichen GDOS-Konfiguration kann über das soeben erwähnte Accessory auch ein in NVDI integrierter Mausbeschleuniger ein- bzw. abgeschaltet werden.

NVDI oder Turbo ST?

Nun zur Gretchenfrage: Lohnt sich die Anschaffung von NVDI? Schließlich gibt es ja auch andere Programme, die für eine schnellere Bildschirmausgabe sorgen. Oder bietet NVDI mehr? Diese Frage läßt sich klar mit ja beantworten. Schließlich darf man nicht vergessen, daß NVDI nicht einfach nur ein schneller Bildschirmtreiber ist, sondern auch ein komplettes VDI inklusive GDOS-System darstellt. (Für Programmierer von Bedeutung: Auch der zusätzliche Befehlsumfang des VDI im GEM V2.x wird unterstützt.) Turbo ST ist nur noch bei Zeichenausgaben über das BIOS schneller als NVDI, nicht jedoch bei der wichtigeren Grafikausgabe über das GEM.

Im Gegensatz zu Turbo ST arbeitet NVDI ohne Probleme mit billigen Grafikerweiterungen wie Overscan oder MegaScreen zusammen. Ebenso werden Großbildschirme wie z.B. die von Matrix und die MAXON MGE-Grafikkarte (hier allerdings nur in der Monochromauflösung) unterstützt.

Für diejenigen, die stets zuwenig freien Speicher besitzen, dürfte noch interessant sein, daß NVDI sparsamer mit Speicherplatz umgeht als Turbo ST.

Alles in allem stellt sich nicht die Frage, ob sich der Kauf von NVDI lohnt (mit 99 DM ist man dabei), sondern es ist eher angebracht, sich Gedanken über die Zukunft von Turbo ST zu machen. Für alle diejenigen, die noch nicht im Besitz von Turbo ST sind, stellt NVDI eindeutig die bessere Alternative dar.

US/HE

Bezugsadresse:

Bela Computer GmbH
Unterortstr. 23-25
6236 Eschborn
Tel.: 06196 (481944)

Quickindex-Geschwindigkeitsvergleich

	TOS 1.4 Turbo ST	TOS 1.4 GDOS V1.1	TOS 1.4 NVDI
Text:	326%	100%	285%
String:	1347%	100%	1069%
Scroll:	136%	100%	131%
Draw:	284%	78%	308%

	KAOS-TOS 1.41	KAOS-TOS NVDI	Atari TT* norm. / fast	TT/NVDI* norm. / fast
Text:	98%	286%	210% / 223%	213% / 635%
String:	117%	1075%	208% / 208%	2787% / 2828%
Scroll:	101%	132%	225% / 225%	237% / 237%
Draw:	114%	367%	180% / 180%	452% / 452%

* Auf dem TT wurde nur im ST-Hoch-Modus im Normal- und im Fast-RAM gemessen.

NVDI-Test

Bei dem NVDI-Test handelt es sich um einen speziellen VDI-Test, der sich auch auf der NVDI-Diskette befindet. Natürlich könnte man jetzt sagen, daß dieser Test "getürkt" sein könnte, aber da er auch genau beschreibt, wie und was er überprüft, und man somit das Ergebnis selbst nachprüfen kann, wollten wir nicht darauf verzichten. Hier kurz zusammengefaßt, was getestet wird:

v_gtext (7, 8, 39 und 40 Zeichen, Höhe 4, 6 und13, Effekt 0)
v_gtext (20 Zeichen, Höhe 26, Effekt 0)
v_gtext (7, 8, 39 und 40 Zeichen, Höhe 4, 6 und13, Effekt 1)
v_gtext (20 Zeichen, Höhe13, Effekt 29)
v_justified (8+40 Zeichen, Höhe 13, Effekt 8)
v_pline (horizontal, vertikal und schräg)
vr_recfl (48*20, 100*100 und 638*400 Punkte, Füllfarbe 0, Füllstil 0, Füllindex 0)
vr_recfl (48*20, 100*100 und 638*400 Punkte, Füllfarbe 1, Füllstil 2, Füllindex 3)
vrt_cpyfm (100*100 und 640*200 Punkte, horizontal und vertikal verschoben, Quelle<Ziel)
vrt_cpyfm (640*200 Punkte, horizontal und vertikal verschoben, Quelle>Ziel)
vro_cpyfm (100*100 und 640*200 Punkte, horizontal und vertikal verschoben, Quelle<Ziel)
vro_cpyfm (640*200 Punkte, horizontal und vertikal verschoben, Quelle>Ziel)
Je 1000 Aufrufe von vst_height, vst_points, vst_color, vst_effects, vst_alignment, vsl_color, vsl_type, vsl_udsty, vsf_interior, vsf_style, vsf_color, vswr_mode, vql_attributes, vqf_attributes, vqt_attributes, vqt_extend, vqt_width, vqt_name, vqt_fontinfo, vqin_mode und vq_color
v_curtext (mit und ohne Scrolling)
Bconout (mit und ohne Scrolling)
Cconws (mit und ohne Scrolling)

Die Ergebnisse

	TOS 1.4 NVDI	TOS 1.4	TOS 1.4 Turbo ST	Atari TT*	Atari TT* NVDI
Textausgabe:	182%	781%	601%	404%	2409%
Linien:	192%	398%	407%	390%	812%
Rechtecke:	518%	768%	792%	498%	1176%
Polygone:	113%	208%	173%	352%	742%
Kreise/Ellipsen:	104%	346%	119%	435%	1212%
Rasteroperationen:	726%	795%	725%	455%	881%
Attributfunktionen:	100%	335%	321%	195%	961%
Auskunftsfunktionen:	100%	280%	239%	213%	829%
Escape-Funktionen:	124%	196%	212%	272%	571%
BIOS-Ausgabe:	118%	201%	228%	234%	434%
GEMDOS-Ausgabe:	111%	631%	731%	222%	1704%
AES-Objekt-Ausgabe:	132%	295%	308%	256%	688%

Alle Messungen wurden, sofern möglich, mit eingeschaltetem Blitter durchgeführt.
* Die Messungen auf dem Atari TT wurden alle in der hohen ST-Auflösung durchgeführt.

Atari MEGA STE



Der große Bruder des 1040STE

Atari sorgt dieses Jahr für eine Weihnachtsüberraschung in der ST-Gemeinde. Zwar schwirrten bereits seit der letzten CeBIT Gerüchte um einen Atari MEGA STE herum. Doch in den Chefetagen hüllte man sich in eisiges Schweigen. Genaues war nicht zu erfahren bzw. nicht für die Ohren der breiten Öffentlichkeit bestimmt. Wenn ich mich recht erinnere, wurde von Sam Tramiel, seines Zeichens Atari-Chef in USA, in einem Interview in einer anderen ST-Zeitschrift sogar der Sinn und Zweck eines solchen Gerätes angezweifelt. Nun, er hat ihn mittlerweile wohl entdeckt.

Leider kann man den neuen MEGA STE mangels Masse noch nicht ausgiebig unter die Lupe nehmen. Daß die neue Maschine von Atari real ist, ist sicher, da sie auf der Comdex-Messe in den USA gezeigt wurde und die wesentlichen Daten, die wir Ihnen auch nicht vorenthalten wollen, aus einem dort verteilten Atari-Prospekt stammen. Genaueres läßt sich wahrscheinlich in der nächsten Ausgabe sagen, da Atari den MEGA STE auf einer Pressekonferenz Mitte Dezember in München vorstellen wird.

Mixed up

Im MEGA STE sind viele Features des Atari TT und des 1040 STE miteinander vereinigt. Was die Sound- und Grafikleistung der Hardware angeht, entspricht der MEGA STE dem 1040 STE, also maximale Auflösung von 640x400 Pixeln, eine Palette von 4096 Farben, hardwaremäßiges Scrolling und programmierbarer Sound-Generator (Stereo). Auch unser alter Freund, der Blitter, bleibt uns erhalten.

Betrachtet man den MEGA STE dagegen rein äußerlich, könnte man ihn für einen atari-grauen TT halten. Das Gehäuse entspricht also dem „modischen“ Design von Ataris Flaggschiff. Sieht man sich das Gerät dann etwas näher an, fällt zumindest auf, daß im Gegensatz zum TT die SCSI-Schnittstelle fehlt.

Beim Prozessor handelt es sich um einen mit 16 MHz getakteten 68000, der auch auf 8 MHz „heruntergeschaltet“ werden kann. Der MEGA STE dürfte somit doppelt so schnell wie ein normaler ST sein und den Beschleunigerkarten für den ST das Leben schwer machen. Er verfügt über einen externen 16-Bit- und einen internen 32-Bit-Bus und kann 24 Bits adressieren. Optional kann man einen Arithmetik-Coprozessor einsetzen, damit auch rechenaufwendige Programme optimal unterstützt werden. Apropos Geschwindigkeit: der MEGA STE verfügt über einen 16-kB-Cache-Speicher, der zusammen mit den 16 MHz ein schnelles Arbeiten ermöglicht. Es wird eine Version mit 2 und eine mit 4 MB RAM geben. Die maximale Ausbaustufe ist 4 MB. Die Aufrüstung auf 4 MB soll laut Prospekt einfach sein, was auf zeitgemäße SIMM-RAMs schließen läßt. Reinstecken, fertig! (Interessant wird sicherlich der Test, ob die 14 MB-Speichererweiterung von Richter funktioniert.)

Im Gegensatz zu den bisherigen ST-Modellen hat der MEGA STE 256 kB ROM (bisher 192 kB). Der zusätzliche Platz wird zum Teil sicherlich von zwei neuen Features benötigt, die wir schon vom TT her kennen: dem neuen Desktop mit Tastaturbefehlen (Short Cuts), Ablegen von Programmen als Icons auf dem Desktop u.v.m. und dem neuen Kontrollfeld, das sich über eine interne Liste erweitern läßt (sofern Atari mal eine Dokumentation herausbringt, wie das geht und was berücksichtigt werden muß).

Kontakte zur Außenwelt

Natürlich braucht der MEGA STE auch Schnittstellen zur Außenwelt. Hier hat man bei Atari nicht gerade gegeizt und einen sehr kompakten Rechner geschaffen. Es befinden sich ein doppelseitiges 3,5"-Diskettenlaufwerk (720 kB) und wahlweise eine 46- oder 80-MB-Festplatte im Gehäuse. Es wird also keine grauen Türme mehr geben, die, bestehend aus Rechner, Festplatten, Wechselplatten, Monitor usw., gen Himmel streben. Atari versucht das Wichtigste in einem Gehäuse unterzubringen. Die Frage ist nur, was ist, wenn eine Festplatte nicht mehr ausreicht. Das Designer-Gehäuse ist nicht gerade zum Stapeln geeignet. Aber vielleicht bringt Atari ja noch eine Tower-Version heraus. Es wäre zumindest wünschenswert, denn zumindest eine Wechselplatte ist sicherlich bei vielen professionellen ST-Besitzern im Einsatz - sei es als Backup-Medium oder auch nur, um große Dateien zu transportieren, was ja bei Bildverarbeitung und DTP leicht passieren kann. Es besteht zwar weiterhin die Möglichkeit, ein zusätzliches Gerät an den DMA-Port (ACSI) anzuschließen, aber die Stapelbarkeit fehlt.

Neben den Atari-üblichen Schnittstellen (MIDI-Ports, externe Floppy, DMA für Festplatte, CD-ROM etc.) gibt es für Audio Out zwei Cinch-Buchsen, eine zweite serielle Schnittstelle (RS232C), einen schnellen DMA-seriellen Port, der auch für ein LAN-Netzwerk benutzt werden kann, und einen eingebauten Modulator für direkten Anschluß von Fernseher oder Videorekorder. Ebenfalls wurde der VME-Port des TT beim MEGA STE eingebaut. Leider ist dadurch der MEGA-Bus der bisherigen MEGA STs weggefallen, wodurch alle Erweiterungskarten, die diesen Bus nutzen, inkompatibel werden.

Neu ist auch der 14"-Farbmonitor SC 1435, der, im TT-Design gehalten, genau auf das Gehäuse paßt. Es läßt sich aber auch weiterhin ein Monochrommonitor anschließen. Ob - wie beim TT - auch ein Großbildschirm direkt angeschlossen werden kann, konnten wir leider noch nicht herausfinden. Auch ein Preis für den deutschen Markt steht noch nicht fest. Wie bereits oben erwähnt, findet Mitte Dezember ja die offizielle Vorstellung des Geräts statt. Wir werden Sie dann auf dem laufenden halten.

HE

Atari MEGA STE

Prozessor:	68000/16 MHz (umschaltbar auf 8 MHz)
Bus:	16-Bit extern, 32-Bit intern, 24-Bit adressierbar
FPU:	optional 68881/82
RAM:	2 oder 4 MB
ROM:	256 kB intern, 128 kB extern (ROM-Port)
Grafikchip:	Blitter
Farbpalette:	4096 Farben
Grafikauflösungen:	640x400 monochrom 640x200 4 Farben 320x200 16 Farben
Hardware-Scrolling:	horizontal/vertikal

programmierbarer Sound-Generator
Stereo-DMA-Sound zum Abspielen digitalisierter Musik

interne 3,5"-Floppy (720 kB)
interne 46 oder 80 MB-Festplatte

Genlock-Anschluß
DMA/LAN-Port
DMA (ACSI)-Port
MIDI-Ports
Stereo-Audio Out (Cinch)
Monitoranschluß
Parallel-Port
2 serielle Ports
externer Floppy-Port
Maus-Joystick-Ports
Modulator für TV/Video
VME-Port

Neuer Desktop wie TT
Neues Kontrollfeld wie TT

Kommentar

Mit dem Erscheinen der MEGA STEs wird mit Sicherheit die bisherige MEGA-Serie abgelöst werden. Was aus der 1 MB-Version wird, wird sicherlich eine der wichtigsten Fragen bleiben, aber wahrscheinlich wird man auf den 1040 STE zurückgreifen müssen. Es scheint, daß Atari seine komplette Produktlinie neu konzipieren will. Was dem zum Opfer fällt, ist noch nicht heraus, aber ein Anfang ist jetzt zu sehen, auch wenn man den neuen MEGA STE erst nächstes Jahr erhalten kann (Tip der Red.: CeBIT '91).

Interessant ist auch der Blick auf die Konkurrenz. Atari versucht, die Angriffe von Apple zu kontern, das kürzlich seine Billigserie vorstellte. Sicher ist auf jeden Fall, daß der MEGA STE mit seinen Features, insbesonderen seinen 16 MHz, dem Billig-Mac Classic deutlich überlegen ist. Leider ist noch kein Preis für den MEGA STE bekannt, doch dürfte er deutlich unter dem der 16 MHz-Rechnern der Konkurrenz liegen. Der Konter scheint gelungen.

ROCKUS









Köln im Herbst



Der 27. Oktober 1990 ist ein kalter und windiger, doch sonniger Herbsttag. Unser Weg führt an diesem Morgen nach Köln, Schauplatz der Orgatec, sozusagen der EDV-Herbstmesse.

Als wir mittags ankommen, ist es leider nur noch kalt und windig, hoffentlich entschädigt uns die Ausstellung. Eigentlich ist die Orgatec eine Büromesse, doch sind mittlerweile die Inhalte der Büroarbeit eine Untergruppe der EDV geworden. Wir tun der Veranstaltung gewiß nicht unrecht, wenn uns Drehstühle und Bleistiftanspitzer nicht interessieren. Es geht in Köln also um Lösungen einer ganzen Reihe von Problemen, die im Büro auftreten. Von der klassischen Textverarbeitung als Ersatz für die Schreibmaschine bis hin zur Auftrags- und Kundenverwaltung, speziell für den KFZ-Betrieb.

Wenngleich drei Hallen offiziell den Peripheriegeräten und Systemkomponenten vorbehalten sind, so darf man nicht das Angebot und den Überblick erwarten, den die CeBIT alljährlich im Frühjahr bringt. Siemens und Hewlett-Packard, zwei große Hersteller, zeigen denn auch keines ihrer Hardware-Produkte als reinen Selbstzweck. Bei ihnen dienen die Gerätschaften lediglich als Träger für Software-Pakete. Kein Taschenrechner bei HP, kein Drucker bei Siemens - der Freak kommt nicht auf seine Kosten.

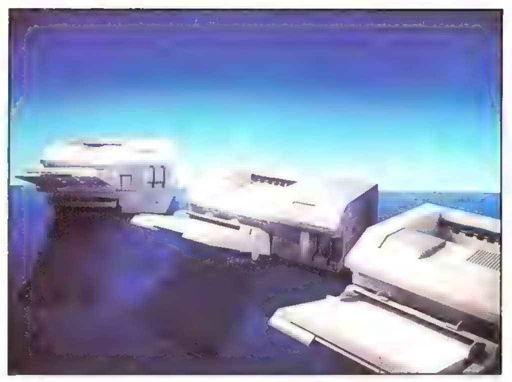
Bei Atari

Doch gerade das macht die Orgatec so interessant. Die Besucher hier haben ganz konkrete Anforderungen. Sie wissen, wozu ihnen der Rechner nutzt oder nutzen soll. Unser erster Weg führt natürlich zum Atari-Stand. Hier zeigen, wie gewohnt, die verschiedenen Software-Hersteller ihre Produkte an den einzelnen Tischen. Die Messe sei ruhig, aber geprägt von der Fachkenntnis der Besucher. Das ist die einhellige Meinung, als wir reihum nach der Stimmung fragen.

Von Atari selbst gab's nur eine klitzekleine Neuigkeit: Der Großbildschirm zum TT ist endlich lieferbar. Der TTM 194 (ECL-Technik), so der Name, ist ein 19"-Monochrommonitor, verfügt über eine Bildauflösung von 1280x960 Bildpunkten,



Der neue TTM 194 von Atari: ein leistungsstarker High-Resolution Monitor für den TT



Schick anzusehen: Die neuen EPL-Laserdrucker von Epson



Ein Kyocera für (fast) jedermann und jederfrau: der F-800 T

arbeitet mit einer Zeilefrequenz von 72 kHz und einer Bildfrequenz von 72 Hz. Der TTM 194 kostet einzeln DM 2198,- und zusammen mit dem TT030/4 DM 8498,-.

Der Blick auf Tempus Word zeigt uns eine Version, die verdammt nah an der 1.0 ist. Die Menüs, die Dialoge, alles ist komplett umgestrickt worden und macht einen sehr guten Eindruck. Das Einbinden von Bildern und die Rechenfunktion, aber auch ein Lexikon mit Duden-Umfang sind gegenüber der von uns in der Juni- Ausgabe getesteten Version noch hinzugekommen. Eine wahre Freude ist es, das Programm auf einem TT zu sehen. Es arbeitet dann trotz grafischer Darstellung schneller als Tempus.

Ebenfalls ein alter Bekannter ist DynaCadd. Sicherlich haben Sie unseren Test in der Oktober-Ausgabe letzten Jahres gelesen. Seitdem hat sich einiges getan in Kanada, woher das Programm kommt. Eine Version für AT- kompatible Rechner ist nun verfügbar, eine solche für den Amiga und eine speziell für den Atari TT stehen kurz vor der Vollendung. Auf der CeBIT soll ebenfalls die Macintosh-Version gezeigt werden. Und - last not least - über eine UNIX-Variante wird ebenfalls nachgedacht. Alle Programme haben den gleichen Preis von 2623 DM. Sicher nicht uninteressant, ein professionelles CAD-System einzusetzen, das auf so vielen Systemen lauffähig ist. Die Krönung wäre dann die uneingeschränkte Netzwerkfähigkeit des Programms, um alle Maschinen z.B. via Ethernet unter einen Hut zu bekommen.

Doch die wichtigste Neuerung ist, daß es wieder einen deutschen Vertrieb für DynaCADD gibt. Die Firma CRP Koruk hat sich der Software hier in Deutschland angenommen. CRP führen auch Updates von allen bisherigen Versionen auf die momentan aktuelle V1.82 durch, die für nicht gerade barmherzige 790 DM nunmehr deutsche Handbücher beinhalten. Ferner werden von der Konstanzer Firma auch Komplettsysteme und Schulungen angeboten.

Kleiner Druckerrundgang

Eine klare Tendenz im Druckermarkt zeichnete sich ja bereits auf der CeBIT ab: Die Laser kommen. Immer preiswerter werden sie, und in dieser Beziehung gibt Atari mit dem neuen SLM 605 mal wieder eine Orientierung (siehe Test in dieser Ausgabe). Wir wollten wissen, was sich sonst noch tut im Markt der schreibenden Ausgabegeräte.

Nicht enttäuscht hat uns Epson. Im Gegensatz zu fast allen anderen Herstellern, die uns auf die CeBIT vertrösteten, gabs hier richtig neue Drucker zu sehen. Der bereits in Hannover gezeigte EPJ 200 ist nun lieferbar. Das 3600 DM teure Gerät ist ein Zwitter: Als Tintenstrahler druckt er aus 64 Düsen, emuliert dabei mit 300 DPI Auflösung jedoch einen Laserdrucker. Das Papierformat beträgt DIN A3, ein automatischer Einzelblatteinzug ist drin und Endlospapier ist ebenso möglich. Epsons neue Laserdrucker-Reihe heißt EPL. Sie beginnt mit dem 7100, der mit 4000 DM der preiswerteste ist. Besonderheit: Der Drucker ist über seine beiden Schnittstellen von zwei Rechnern aus gleichzeitig anzusprechen. Die Reihe endet mit dem 7500, der eine PostScript-Emulation in seinem Innern birgt.

Kyocera-Produkte decken mehr den High-End- und Profi-Bereich ab. Doch von dort kommt eine neue Reihe von Laserdruckern, die sich durch die Möglichkeit auszeichnet, den Ausbau der Geräte aufzustocken. Der kleinste ist der F-800 T, der mit sage und schreibe 79 festen Fonts und drei skalierbaren daherkommt. Er kostet 4218 DM. Je nach Speicherausbau verspricht uns Kyocera die Eignung für besondere Einsätze: DTP, CAD. Dann bekommen die Kinder natürlich auch andere Namen und heißen dann F-800 TI und TIV.

Von Minolta kennen wir eigentlich die Kameras und (wir sind nicht auf der Photokina!) noch die Kopierer. Nun will auch dieser Hersteller sein Know-How nicht einzig den OEM-Partnern zur Verfügung stellen. Es gibt also auch einen Laserdrucker von Minolta, den SP 101, der preislich unter 4000 DM liegt. Mit 32 Fonts und einen speziellen Fein-Toning System kommt er auf den Schreibtisch.

Allgemeine Information zu NEC PC Engine

NEC Corporation und NEC Home Electronics Ltd., Tokio, (nachfolgend „NEC“ genannt) informieren hiermit sämtliche Benutzer der NEC PC Engine-Geräte Super GrafX und Core GrafX.

1. Die NEC PC Engine-Geräte Super GrafX und Core GrafX werden von der Firma NEC Home Electronics Ltd. für Übertragungssysteme, die nicht mit dem deutschen Fernsehübertragungssystem verwendet werden können, hergestellt.

Aus diesem Grund vertreibt NEC die PC Engine-Geräte weder in Deutschland noch in anderen EG-Staaten. Daher werden sämtliche NEC PC Engine-Geräte in Deutschland weder direkt noch indirekt von NEC, sondern ausschließlich von Drittfirmen zum Kauf bzw. zur Verwendung angeboten.

2. NEC hat kürzlich davon Kenntnis erhalten, daß gegenwärtig durch nicht von NEC autorisierte Vertriebsfirmen und Händler in Deutschland geänderte PC Engine-Geräte verkauft werden.

Im Hinblick auf derartige Änderungen der NEC Geräte sowie deren Verkauf und Verwendung übernimmt NEC keinerlei Haftung oder Gewährleistung im Falle von Fehlern oder Unfällen bei der Verwendung dieser ungenehmigten Geräte.

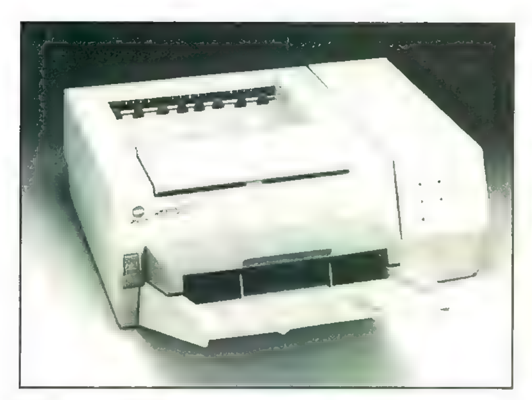
3. Für Fehler an in Deutschland verkauften, geänderten oder nicht geänderten PC Engine-Geräten ist NEC weder verantwortlich noch bereit, in irgendwelche Verhandlungen zwischen Käufern, Benutzern und nicht autorisierten Vertriebsfirmen und Händlern einbezogen zu werden.

Insbesondere kann NEC keine Serviceleistungen für geänderte oder nicht geänderte PC Engine-Geräte in Deutschland oder anderswo durchführen. Sofern derartige Geräte zu NEC mit der Bitte um Überprüfung oder für Servicearbeiten übersandt werden, wird NEC keinerlei Serviceunterstützung gewähren.

Die Garantieerklärungen für PC Engine-Geräte, die in Japan verkauft werden, beziehen sich nur auf nicht geänderte Geräte, welche in Japan verwendet werden.

Etwaige Garantieerklärungen von nicht autorisierten Händlern oder Vertriebsfirmen für geänderte oder nicht geänderte PC Engine-Geräte in Deutschland, berühren NEC Garantieverpflichtungen nicht.

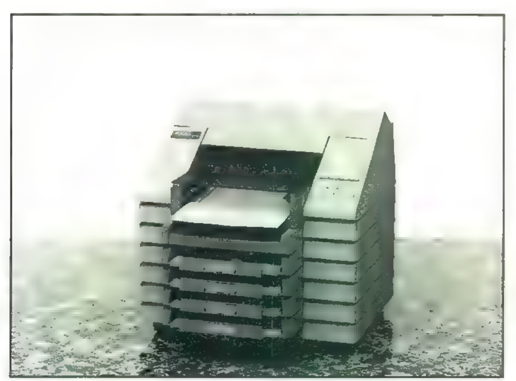




Ein Laser vom Kopiererspezialisten: der Minolta SP 101



Drei auf einen Streich: neue 9- und 24-Nadler von Star



Kein Problem mit vielen Formularen: Multischachtdrucker von BDT

Hochstapler

Hoch hinaus will der deutsche Hersteller BDT mit seinen Druk-kern. Die fallen bereits optisch aus dem Rahmen, denn sie verfügen nicht nur über einen oder zwei Einzelblattschächte. Man ist in Rottweil nicht kleinlich und fängt mit vieren an. Das verspricht bestes Papier-Handling überall dort, wo mit verschiedenen Formularen gearbeitet wird. Die Maske ist schnell in die Textverarbeitung eingegeben, doch vor dem Druck muß noch das richtige Formular eingespannt werden. Der neue Ergoprint 310 von BDT setzt dem nun ein Ende. Vier oder sechs Papierzuführungen, deren Lage der Drucker erkennt (man kann die Kassetten also vertauschen), rationalisieren die Arbeit in Praxen und Büros. Auf dem Bild ist unschwer zu erkennen, daß es sich bei der Maschine selbst um einen modifizierten HP-Deskjet handelt. Mit 300 DPI und Tintenstrahltechnik ein qualitativ hochwertiges und leises Druckwerk.

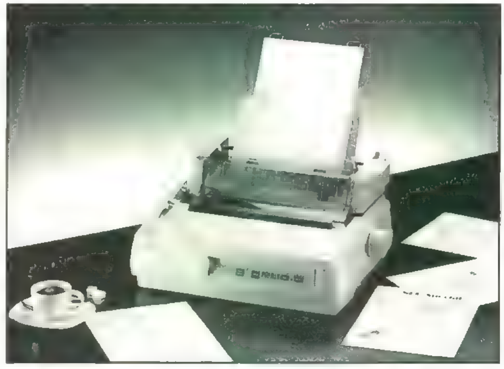
Es gibt noch Nadeln

Bei einigen Ausstellern haben wir den Eindruck, als wollten sie ganz und gar die Laserdrucker in den Vordergrund stellen. Bei Star ist das anders. Hier liegt nahezu das ganze Geschäft auf den pieksigen Rabauken. Demnach gibt's auch gleich Neuigkeiten: Drei Drucker, die alle in das Marktsegment passen, in dem sich Star am wohlsten fühlt: Am unteren Preisrand nämlich. Der LC-20 ist der Nachfolger des LC-10, des bislang billigsten und erfolgreichsten 9-Nadlers von Star. Der unverbindliche Preis liegt bei 550 DM - mal sehen, was der Handel draus macht. Er ist ein wenig schneller, hat vier Fonts und nun auch mechanische Tasten. Sein Gehäuse, wie auch das seiner Brüder - zeigt modische Kurven im Windkanal-Look. Der 750 DM teure LC-200 ist eine ähnliche Maschine. Ebenfalls neunnadelig, verarbeitet er Blätter in DIN A3-Breite, druckt jedoch nur DIN A4. Besonderheit: Das Papier kann nun auch durch den Gehäuseboden zugeführt werden, der Schubtraktor (mit Paper-Park, logo) mutiert dafür zum Zugtraktor. Und ein 24-Nadler darf natürlich nicht fehlen. Der LC24-200 bedruckt DIN A3-Format, kann das Papier ebenfalls von unten einziehen und glänzt mit vielen Schriften. Er kostet 1000 DM, die Farbversion mit dem Zusatz 'CL' im Namen noch einmal 100 DM mehr.

Und noch mehr Nadeln: Citizen, mit dem Swift24 (siehe Test in der Mai-Ausgabe) ja sehr erfolgreich, hat das Gerät innerhalb eines Jahres 75000 mal in Europa verkauft. Jetzt ist es abgespeckt worden und kommt als 'Abrundung nach unten'. Die neue Maschine heißt '124D'. 2 LQ-Fonts, Papier-Park, NEC P6-Emulation, das sind die Features des 798 DM teuren Einstiegs-24-Nadlers. Offenbar ist die Nachfrage aus dem betrieblichen Bereich groß, denn nun gibt es auch eine breite, eine DIN A3-Version des Swift24 mit dem Namen Swift24X und dem Preis von 1598 DM.

Wiederum im ganz unteren Preisbereich setzt Seikoshas neuestes Produkt an: Der SP-1900 AI ist mit 499 einer der billigsten 9-Nadler überhaupt. Er soll vor allem über Kaufhäuser verkauft werden, und dann wird der Preis sicher noch drunter liegen. Dabei ist er mit Epson- und IBM-Emulation, Papier-Park und halbautomatischem Einzug nicht einmal mager ausgestattet.

Für besondere Aufmerksamkeit sorgen die neuen Produkte von NEC. Da sind zunächst der P20 und P30 zu nennen. Die beiden unterscheiden sich lediglich in der Druckbreite. Sie lösen den bisherigen Low-Cost-24-Nadler von NEC ab, den P2plus. Dabei



Ablösung für den P2plus: Der P20 ist der neue kleine NEC.



Leisetreter: Der Silentwriter2 S60 ist NECs Low-Cost-Laser.

ähneln sie äußerlich und innerlich stark dem P60/70 (siehe Test in der September-Ausgabe). Sie sind lediglich langsamer und nicht ganz so komfortabel. Hinzugekommen ist aber ein 'Flip'-Traktor, der sich von Zug- auf Schubbetrieb umbauen läßt.

Noch mehr Laser

Neu bei NEC sind ebenfalls die Laser mit den schönen Namen Silentwriter2 S60 und S60P. Die beiden unterscheiden sich durch die PostScript-Fähigkeit des letztgenannten. Sie basieren - wie übrigens die neuen Epson-Laser auch - auf dem Druckwerk des bereits erwähnten Minolta SP 101. Bei NEC ist man jedoch stolz auf den speziellen Controller, den man den Geräten eingepflanzt hat. Sie sollen schnell sein und den Toner äußerst effektiv nutzen. In der Standardausführung besitzen beide Maschinen 1,5 bzw. 2 MByte Speicher. Mit der Erweiterung desselbigen kann man sich also getrost Zeit lassen. Der 'normale' S60 ist kompatibel zum HP-LaserJet und wird unter 4000 DM kosten.

Eine ganz schlaue Methode, um sich die Speichererweiterei bei den Laserdruckern zu ersparen, ist den Brother-Technikern eingefallen. Diese Erweiterungen sind nötig, weil ein Laser- oder besser: 'Seitendrucker' seitenorientiert arbeitet und diese Seiten komplett im Speicher aufbereiten muß, ehe er sie druckt. Wenn es sich dabei um ein Bild handelt, braucht er dafür ein Megabyte Platz. Doch lassen sich gerade Bilddaten phantastisch komprimieren. Das sieht man ja an allen komprimierenden Dateiformaten (*.IMG, *.TIF etc.). Warum sollte also nicht auch ein Drucker in der Lage sein, die Daten komprimiert im Speicher abzulegen und erst beim Druckvorgang 'auszupacken'? Genau das tun fortan alle Brother-Seitendrucker. Vom kleinen HL-4 bis zum großen HL-8PS. Und jetzt sollen auch für eine komplette Grafikseite 512 kByte reichen. Der HL-4 hat übrigens ein Brüderchen bekommen: Der HL-4 PS verfügt über eine PostScript-Emulation, genannt BrotherScript.

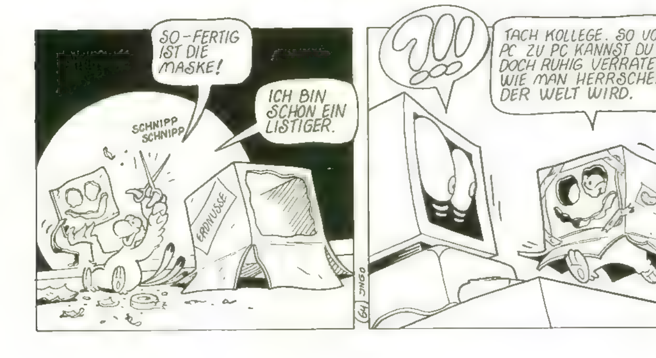
Nicht unerwähnt bleiben sollen zwei ganz unterschiedliche Maschinen, die eines gemeinsam haben: die Tintenstrahltechnik. Canon hat da mit dem Bubblejet bereits Erfahrung. Der Neue zielt aber direkt in den Aktenkoffer. Er heißt BJ-10e, ist so groß wie ein DIN A4-Blatt, ca 4 cm hoch und wiegt ganze 1,8 kg. Somit hat sich der kleine, aber bereits wichtige Markt spezieller Laptop-Drucker weiter verdichtet.

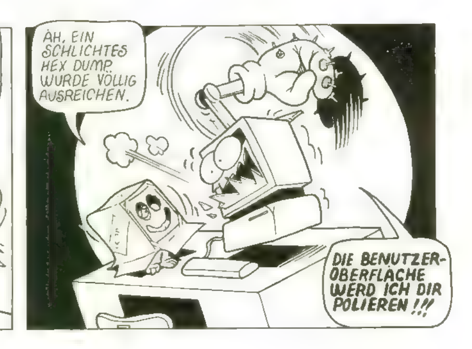
Mit dem Diconix, einem kleinen Tintenstrahler, ebenfalls für den Laptop-Bereich, hat auch Kodak einen Fuß in diesem Markt. Nun gibt es den Diconix 330C, wobei das 'C' für Farbe steht. Er emuliert den HP PaintJet, den wohl verbreitetsten Farb-Tintenstrahler. Ab nächstem Jahr soll der neue Kodak für 3800 DM erhältlich sein.

Hiermit beenden wir unseren Rundgang über die Orgatec'90. Natürlich nicht, ohne Ihnen versprochen zu haben, daß Sie die interessantesten der hier vorgestellten Geräte demnächst in diesem Theater erleben.

IB







Alles Lesens-Werte

Experimente am Schreibtisch

Durch schnelle Computer kann seit geraumer Zeit die Simulation von Naturvorgängen realistisch vorgeführt werden.
Mit diesem Buch werden Sie in die Lage versetzt, in Ihrem ATARI ST Galaxien kollidieren zu lassen, ohne daß der Himmel einstürzt, oder gar die gewagtesten chemischen Experimente zu riskieren, ohne daß gleich das Haus in die Luft gesprengt wird.
Natürlich darf bei allem Spieltrieb die Theorie der Gesetzmäßigkeiten nicht zu kurz kommen. Ausführlich dokumentierte Listings in GFA-BASIC geben einen Einblick in die programmiertechnische Seite.



Robert Becker
Computersimulationen
337 Seiten und Diskette
DM 59,-
ISBN 3-927065-03-X

Mehr als ein Buch! Mehr als nur Software!

2. überarbeitete Auflage ab Ende Dezember erhältlich

Mittlerweile zählt Scheibenkleister II bereits zu den Standardwerken für den ST. Die Erfolgsautoren Claus Brod und Anton Stepper beschreiben auf fast 900 Seiten alles, was man über Floppies, Festplatten, CD-ROMs und andere Massenspeicher wissen muß. Das Buch beinhaltet einen großen Kursteil, in dem Themen wie Floppy- und Festplattenprogrammierung (über Betriebssystem und direkte Controllerprogrammierung), Aufzeichnungsverfahren, Anschluß von Fremdlaufwerken und sogar Justierung und Reparaturhinweise von Laufwerken u.v.m. Zusätzlich wird eine Diskette mit 1.2 MB Software mitgeliefert, die keine Wünsche offen läßt:
Track- und Sektormonitor, bootfähige RAM-Disk, Hyperformat bringt bis zu 950 kByte, Festplattentreiber CBHD.SYS (Laden von Accessories von beliebigem Laufwerk) u.v.m.



Brod/Stepper
Scheibenkleister II
872 Seiten
und Diskette
DM 89,-
ISBN 3-927065-00-5

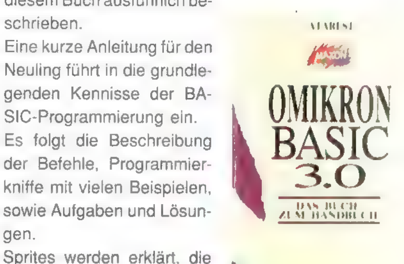
Fürs Programmieren unentbehrlich

Die Art und Weise wie benutzererstellte Daten behandelt werden, ob beim Speichern, beim Sortieren oder beim Suchen, ist ein zentrales Thema in der Informatik. Grund genug für Dirk Brockhaus, den verschiedenen Strukturen und Algorithmen auf mehr als 400 Seiten auf den Leib zu rücken.
Angefangen mit den Grundlagen, wie einfachste Strukturelemente, über komplexe Zahlen, sowie verschiedene Methoden für Suchen und Sortieren, bis hin zur Verarbeitung großer Datenmengen umspannt dieses Buch den gesamten Themenbereich.

Dirk Brockhaus
Datenstrukturen
403 Seiten und Diskette
DM 59,-
ISBN 3-927065-02-1

Das Buch zum Handbuch

OMIKRON.BASIC, mittlerweile als ATARI-BASIC bei jedem ST beigefügt, wird in diesem Buch ausführlich beschrieben.
Eine kurze Anleitung für den Neuling führt in die grundlegenden Kennisse der BASIC-Programmierung ein.
Es folgt die Beschreibung der Befehle, Programmierkniffe mit vielen Beispielen, sowie Aufgaben und Lösungen.
Sprites werden erklärt, die Overlaytechnik zeigt das Auslagern langer Programmteile nebst Laden bei Gebrauch.
Was sind Bibliotheken (Libraries)? Das Mitbenutzen fremder Bibliotheksfunktionen, Aufrufe von TOS und GEM, sowie GDOS wird kein Geheimnis bleiben.



Ein weiterer Schwerpunkt ist die Sound- und Grafikprogrammierung unter OMIKRON-Basic, mit Klarheit über Metafiles, IMG-Format usw. Als Spezialität folgt die Beschreibung von internen Multitasking-Befehlen.

Clemens Hoffmann
OMIKRON.BASIC 3.0
355 Seiten
und Diskette
DM 59,-
ISBN 3-927065-01-3





Bestellcoupon MAXON Computer GmbH Schwalbacherstr. 52 6236 Eschborn Tel.: 06196/481811

Name/Vorname ____________________

Straße ____________________

Ort ____________________

Unterschrift ____________________

Hiermit bestelle ich:

- ☐ SCHEIBENKLEISTER II DM 89,00
- ☐ COMPUTER-SIMULATIONEN DM 59,00
- ☐ OMIKRON.BASIC 3.0 DM 59,00
- ☐ DATENSTRUKTUREN DM 59,00

Versandkosten: Inland DM 7,50
Ausland DM 10,00

Auslandbestellungen **nur** gegen Vorauskasse
Nachnahme zuzgl. DM 4,00 Nachnahmegebühr.

☐ Vorauskasse
☐ Nachnahme

Vertrieb in der Schweiz: DTZ DataTrade AG Langstrasse 94 Postfach 413 CH-8021 Zurich Tel · 01/242 80 88 Fax · 01/291 05 07
Vertrieb in Österreich: Dipl.-Ing. Reinhart Temmel Ges.m.b.H. & Co.KG Markt 109 A-5440 Golling Tel. 06244/7081 17 Fax.: 06244/7188-3
Vertrieb in Frankreich: AROBACE 2 Rue Piemontesis F-75018 Paris Tel: 1/42235044 Fax 1 42545631

ARTWORKS-Collection

Calamus nimmt Gestalt an



Vor etwa 10 Jahren hat sie stattgefunden, die stille Revolution in den Schreibstuben der Zeitungsredaktionen. Hand-Layouter und Maschinensetzer sahen mit Recht ihren Arbeitsplatz gefährdet, als ein neues Hilfsmittel, ein Teufelswerkzeug, die Macht übernahm: der Computersatz.

Redakteure tippten die Texte in Terminals, und aus dem Belichter kam der fertige Satzumbruch heraus. Und das war erst der Anfang!

Dann wurden die Gestalterarbeitsplätze mit ihren altmodischen Durchleuchterplatten ebenfalls durch Terminals abgelöst. Selbst die Rotationswalzen wurden alsbald per Laserstrahl „beschrieben", so daß auch dort die Handarbeit völlig entfiel. Der Weg vom Autor bis in die Rotation ist fast schon ein lückenlos elektronischer geworden.

Durch schnelle Computer, fortschrittliche Bedienerführung (Window-Technik), gute hochauflösende Grafikmonitore und entsprechend leistungsfähige Software propagiert die EDV-Industrie seit mehreren Jahren schon eine Anwendung, die es dem PC-Nutzer ermöglicht, Druckwerke höchstselbst an seinem Bildschirm zu entwickeln, ohne die Dienste einer Lithografieanstalt oder einer Offsetdruckerei beanspruchen zu müssen: Desktop Publishing, kurz „DTP".

Es sollte gewissermaßen die Druckerei im Kleinen werden, alles machbar in nur einem einzigen Personal Computer. Ein weiteres Zauberwort machte die Runde: WYSIWYG, what you see is what you get, zu deutsch: Was du siehst, ist was du bekommst. Es soll eine neue Darstellungsweise des Textes auf dem Bildschirm beschreiben: So wie man es sieht, erscheint es auch auf dem Papier.

Alle die gerade beschriebenen Fortschritte in der Computertechnik ebneten dieser neuen Technik den Weg - und wo bleibt der Benutzer?

Sehr schnell ist auch die Firma ATARI auf diesen DTP-Zug aufgesprungen. Beweis: Die sogenannten DTP-Center von ATARI, die durch einen ausgewählten Fachhandel bedient werden. Dort gibt es neben der ganzen Palette an nötiger Hardware auch Software und Schulung, um letztendlich den Traum von der eigenen Druckerei im Kleinen bei dem interessierten Anwender Wirklichkeit werden zu lassen.

Obwohl es schon mindestens drei andere Programme auf dem Markt gab (Timeworks Publisher, BeckerPage, Publisher ST), wurde ganz offen eines für diesen Zweck von der Firma ATARI favorisiert: CALAMUS von der Firma DMC, Walluf. So war CALAMUS bei seinem Debüt 1987 noch nicht absturzsicher und hatte viele der versprochenen Funktionen leider noch nicht eingebaut. Eine Fertigstellung des ursprünglich zugesagten Funktionsumfangs zog sich in die Monate, bis man eine abgespeckte Version doch noch in den Verkauf brachte. Das eigentliche Super-DTP-Calamus läßt noch immer auf sich warten.

Wie dem auch sei: Für viele Anwender ist das, was nunmehr zur Verfügung steht, völlig ausreichend. Ja, für einige Anwender ist der Funktionsumfang sogar schon wieder zuviel (es kommt halt immer darauf an, was man daraus machen will). CALAMUS macht es seinen Nutzern auch nicht gerade leicht, ein fertiges Dokument herauszubekommen. Da sind ellenlange Menüs und verwirrend viele (wenn auch sehr aussagekräftige) Symbolknöpfe (genannt: Icons). Und wer zum ersten Mal vor seinem Bildschirm saß und mit CALAMUS herumprobierte, der wird angesichts der vielen Rahmen, Ziehknöpfe und Schalter sehr schnell die Verwirklichung einer Formularidee ins Reich der Fabel verdammt haben.

Das Problem ist schnell erkannt: Viele DTP-Neulinge sind von der Vielfalt solcher Programme geradezu erschlagen. Niemand sagt ihnen, welchen Rahmen sie wohin legen, welchen Schrifttyp sie wofür

benutzen und welche Schalter und Funktionen sie in welcher Reihenfolge ausführen sollen, vom theoretischen Grundwissen ganz zu schweigen. Ganz bestimmt gibt es irgendwo im weiten Lande einige Schulungstermine für DTP - aber was nützt es, wenn in weiter Ferne jemand vor seinem CALAMUS sitzt und die Funktionsvielfalt bewundert. Es zeigt sich immer wieder, daß ein Programm nur so vielfältig sein kann wie das, was sein Benutzer daraus macht. Und genau hier liegt das Problem.

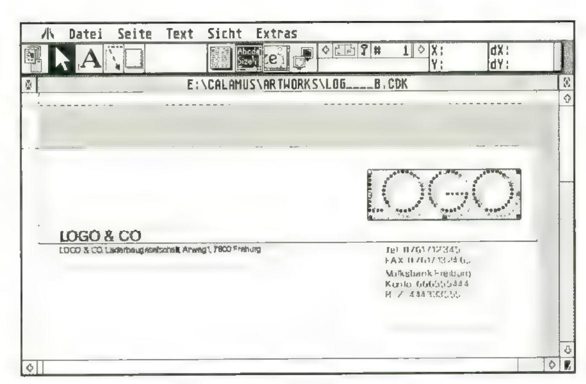


Bild 1: Hier wird ein Briefkopf in Calamus konstruiert.

Artworks
Der andere Weg

Genau dieses Problem haben einige Leute (wohl auch aus leidvoller eigener Erfahrung) erkannt und für Abhilfe gesorgt: Die Artworks-Collection. Die Firma Publishing Design, Hesse und Herwig GbR, versucht mit der Artworks-Reihe einen anderen Weg zu gehen, gewissermaßen das Pferd von hinten aufzuzäumen. Was nützt es einem Benutzer, kiloschwere Handbücher zu wälzen, DTP-Kurse zu besuchen oder gar einen EDV-Berater zu engagieren, im Endeffekt möchte er doch nur eines: seine kleinen Problemchen mittels Computer gelöst zu bekommen, und hierfür hat er sich ein DTP-Programm angeschafft.

So dachten sich die Leute der Artworks-Collection: Wir zeigen nicht vom ersten mühsamen Strich leidvoll Schritt für Schritt, was der Benutzer tun soll, wir liefern ihm gleich die kompletten Resultate fix und fertig. So ist das erste Gestaltungspaket entstanden mit dem Namen Artworks Business. Es will nicht unbedingt nur ein bloßes Werkzeug zur Druckvorlagenerstellung sein, sondern versucht, den Benutzer durch die fertigen Ergebnisse zu fesseln und darauf seine eigene Kreativität aufzubauen.

Artworks ist also eine Sammlung, ein Ideenlieferant bzw. die Grundlage für weitere Ideen der Benutzer.

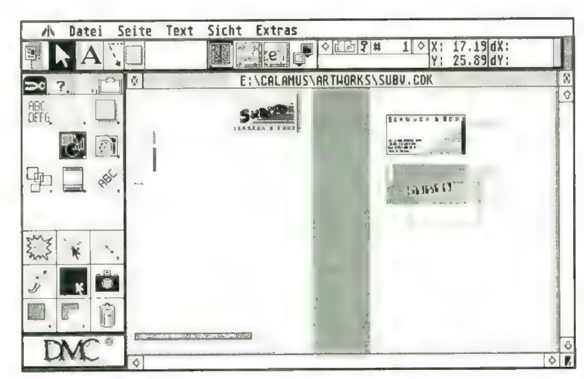


Bild 2: So präsentiert Calamus zwei Vorlagen im Gesamtüberblick.

Artworks
ist kein Programm

Als ich zum ersten Mal die drei Disketten der Artworks-Collection einlud wunderte ich mich. Kein gewohntes *PRG* oder *TTP*, alle Dateien trugen nur die Erweiterung *CDK*. Damit war klar, Artworks ist kein Programm, sondern besteht nur aus CALAMUS-Dokumenten. Damit ist etwas anderes aber ebenso klar: Um die Beispiele aus Artworks anwenden zu können, muß man im Besitz einer CALAMUS-Version sein, sonst läuft nichts.

In der Artworks-Collection sind in der Hauptsache Vorlagen für Briefbögen, Geschäftsformulare (z.B. Kurzbrief, Rechnung, Lieferschein), Visitenkarten, Aufkleber und Inserate zusammengestellt. Diese liegen durchgängig im normalen DIN-A4-Format (210 * 297 mm) und im etwas größeren Linotype-Belichterformat (240 * 327 mm) vor.

Das Problem der CALAMUS-Ausdrucke über den vorgesehenen ATARI-Laserdrucker SLM 804 liegt darin, daß die schrägen Linienverläufe leider noch in einem „Treppenmuster“ herauskommen. Diese Ausdrucke sind als Reprovorlagen nicht sehr gut geeignet, dafür eignet sich die Übergabe der Datei direkt in den Linotype-Belichter besser.

Wenn man sich nun heranwagt, aus dem Füllhorn der Vorlagen eines auf die eigenen Verhältnisse zurechtzubasteln (Namen und Anschrift ändern, evtl. andere Schriftart wählen usw.), dann sollte man schon mit den elementaren Funktionen und der Bedienung von CALAMUS vertraut sein. Zusätzlich kann es nötig sein, noch diverse Hilfsprogramme und Font-Editoren zusätzlich zu Hilfe zu nehmen (z.B. Outline Art).

Alle in Artworks dargestellten Gestaltungsmuster und Schriftzüge sind copyright-frei und dürfen nach Belieben weiterverwendet werden (worauf wir später noch einmal zu sprechen kommen).

Probieren geht
über Studieren

Nun sind wir also beim eigentlichen Sinn und Zweck der Artworks-Collection angelangt. Der Kunde soll in freien Stücken vorhandene Vorlagen so verändern, erweitern, ergänzen, bis sie eine Gestalt angenommen haben, wie sie ihm gefällt.

Also los geht's: CALAMUS laden, ein beliebiges Dokument laden und frei Rahmen verschieben. Bei den sogenannten beweglichen Texten wird das schon etwas schwieriger. Bewegliche Texte sind Angaben wie das Anschriftenfeld, Bankverbindung, Absenderangabe usw. Dort

muß der Rahmen erst selektiert sein, dann geht's in den Texteditor, der alte Text wird gelöscht und der neue geschrieben, aber: der bestehende Schrifttyp (Font) bleibt beibehalten. Natürlich kann der Font ebenfalls nach freiem Belieben getauscht werden.

Wenn Sie, liebe Anwender an diesem Punkt angelangt sind, werden Sie allermeistens mit dem Ergebnis Ihrer Arbeit unzufrieden sein. Obwohl Ihnen ein reicher Schatz an Vorlagen geliefert wurde, scheint Ihnen das, was Ihre eigene Kreativität daraus gemacht hat, nicht sehr ansehnlich zu sein. Das wußten auch die Leute der Artworks-Collection. Deswegen besteht das Paket nicht nur aus drei Disketten mit unzähligen Mustern für alles Mögliche, sondern es ist ein erstaunlich umfangreiches Grundwerk der DTP-Technik als Lehrtext dabei.

DTP-Theoretisches

Auf 88 Seiten wird sehr ausführlich das Grundwissen vermittelt, das nötig ist, um die Entstehung der Formulare zu verstehen. So wird Ihnen nicht unbedingt beigebracht, wie man eine Zeitung oder ein Buch fabriziert, sondern alle Erläuterungen sind auf das Umfeld der geschäftlichen Druckstücke beschränkt. Das will aber nicht heißen, daß nur speziell auf diesen Anwendungszweck abgestellt wurde. Kenntnisse über Layout und Typografie, über Schrifttechnik und Gestaltung können für andere Zwecke dennoch sehr nützlich sein („was man im Kopfe hat...", kleiner Scherz des Autors).

Im Kapitel 2 des Begleitbuches habe ich den wichtigen Satz gelernt: „Wenn der textliche Inhalt, also die Information im Vordergrund steht, dann sollten sich Layout und typografische Stilmittel diesem unterordnen". Das ist sozusagen auch das Grundgesetz des Desktop Publishings. Es kommt also immer darauf an, was man mit dem Dokument aussagen will. Ein Bestattungsunternehmen wird sicher kaum peppige Schriftzüge und auffällige Logos verwenden. Demgegenüber wäre für ein Layout-Studio oder einen Werbegrafiker ein bescheiden einfacher Schriftzug und ein schlichtes Ganzes kaum die richtige Werbung.

Sehr schön ist die Erklärung des Unterschieds zwischen Typografie (Gestaltung von Buchstaben und Schriften) und Layout (Zusammensetzung von Absätzen und Seiten), die da lautet: „Das was Sie in der Ganzseitendarstellung aus großer Entfernung im Dokument noch unterscheiden können, das ist Layout. Was Sie aber in der vergrößerten Darstellung vom Text erkennen, das betrifft die Typografie." (Zitat)

Als nächstes kommt die Wahl der richtigen Schrift. Und das ist gar nicht so einfach, wie Sie denken! Heute sind hauptsächlich zwei Schriftenfamilien im Gebrauch. Die erste heißt „Antiqua". Ihr Merkmal sind die „Serifen", also die leichten Ausschweifungen an den Enden. Sie gehen vom Buchstabenelement oder Grundstrich (Hauptteil des Buchstabens) leicht gerundet (Schrift „Times" oder „Garamond") oder ein wenig eckig bis winkelig (Schrift „Bondi") bis zur Endung weg. Diese Schrift ist heute noch in vielen Büchern und Zeitungen vorherrschend. Quasi das Gegenteil davon ist die zweite Schriftfamilie, genannt „Grotesk". Sie verzichtet völlig auf Schnörkeleien (Schrift „Swiss" oder „Avant Garde"). Sie finden hauptsächlich in Anzeigen, Handzetteln oder Prospekten ihre Anwendung.

Eine dritte Schriftenfamilie findet zunehmend Verbreitung im DTP-Bereich, man nennt sie „Headline". Dort sind meist nur Abwandlungen schon bestehender Schriften zu finden, die stark ins Grafische, Handschriftliche übergehen. Die uralte Diskussion, ob Serifen oder Groteske leichter zu lesen sind, läßt sich nicht abschließend beantworten, es gibt nur einige Faustregeln: Serifen eignen sich für längere Texte, wie in Büchern oder Illustrierten, wobei Groteske mehr für „kurzlebige" Dokumente angenehmer sind.

Nach dieser Einführung kommt der Artworks-Kursus auf Schriften im DTP zu sprechen. Dort wird die Maßeinheit „Punkt" vorgestellt, die auf den französischen Schriftgießer F. Didot zurückgeführt wird. Ein Punkt ist 0,376 mm. Ein guter Lesetext sollte zwischen 9 und 12 Punkt groß sein.

Das Kapitel 3 zeigt nun die Gestaltungselemente von typischen Geschäftsdokumenten wie Briefbogen, Visitenkarte oder Gebrauchsformular. Dort werden Begriffe wie Format, Faltmarken, Lochmarke, Bezugsfeld und Adreßfenster erklärt. Was ist wichtig im Zusammenhang zwischen Adreßfenster, Logo, Farbe und Schrift?

Die Gestaltungspraxis (Kapitel 4) bringt uns die Entwicklungsschritte näher, die nötig sind, um ein vernünftiges Dokument zu erzielen. Es reicht also nicht, sich einfach hinzusetzen und loszulegen. Es muß erst ein sinnvolles Konzept erarbeitet sein, vielleicht unter Zuhilfenahme der bereits vorhandenen Artworks-Muster. So ist dieses Kapitel durchgehend vom Entwurf (am Anfang war das Logo) bis zu Fertigstellung an zwei konkreten Fällen aus der Praxis angelehnt.

Ein kleinwenig später erfährt der Leser, was eine Nutzenerstellung ist und wofür man sie braucht. Es handelt sich am Beispiel der Visitenkarte um die sinnvolle Aufteilung des Papierbogens (wahrscheinlich DIN A4), so daß möglichst viele Visitenkarten darauf Platz finden. Und für jede Druckvariation, also Vorder- und Rückseite, oder auch für jede Farbe, muß ein eigener Nutzen angefertigt werden.

Damit sind wir auch schon im Kapitel 5. Dort finden die hauptsächlichen Druckverfahren ihre Erläuterung. Der spätere Klein-Layouter soll nämlich die Vorarbeiten kennen, die zu Offset- und Siebdruck nötig sind. Dabei spielen einerseits die gewünschte Auflage (Massendruck-

sachen oder Handzettel erledigt man günstiger im Offset), andererseits das Druckergebnis (Aufkleber sind besser im Siebdruck zu erledigen) eine Rolle.

Kapitel 6 (Gestaltungspraxis 2) und 7 (Dokumentengestaltung) führen dann in etwas komplizierte Gefilde. Dort werden umfangreiche Formulare erklärt. Diese Kapitel sind stark an die auf Diskette vorhandenen Beispiele angelehnt, wobei diese je nach Schwierigkeitsgrad eingehend beschrieben sowie deren typische Unterschiede und Besonderheiten dargestellt sind.

Was der Laser alles kann

Kapitel 8 führt gewissermaßen zur logischen Konsequenz der ganzen Vorarbeit: der endgültigen Aufbereitung zum Druck und der Präsentation (auf dem Laserdrucker). Wie wird das heute noch gemacht, ohne DTP-Verfahren? Nach einem Rohentwurf setzt sich der Grafiker hin und bastelt ein „Scribble" zusammen. Dieser Begriff kommt von dem im Computerzeitalter antiquarisch anmutenden „Rubbeln" der Buchstaben von der Folie. Eine Reprokamera bringt das in die richtige Größe. Dann wird geschnippelt und geklebt und das ganze wieder unter die Reprokamera gelegt. Das Ergebnis kann sich ganz bestimmt sehen lassen, so wurde schließlich jahrzehntelang gearbeitet (und nicht einmal schlecht). Doch mit einem Laserdrucker, und der gehört schließlich zur Grundausstattung eines neuen DTP-Arbeitsplatzes, läßt sich schneller und flexibler arbeiten.

Wußten Sie, daß man mit unserem guten SLM 804-Laserdrucker auch mehrfarbige Ausdrucke erzielen kann? Ich wußte es auch nicht und war extrem überrascht, welche einfache Lösung sich hierfür anbot. Stellen Sie sich vor, die tollsten Autoaufkleber aus unserem ATARI-Laser! Es geht wahrhaftig. Ja selbst das Bedrucken von dickerer PVC-Folie oder Namensschildern und Ausweiskarten ist mit dem ATARI-Laser möglich (hierzu ist allerdings ein Zusatzgerät im Preis eines 9-Nadeldruckers nötig).

Mit 88 Seiten ist das dicke Artworks-Handbuch mit dem Theorietitel zwar am Ende, aber es schließen sich noch 55 Blatt mit (teilweise mehrfarbigen) Musterausdrucken an - alle, wie sie auf Disketten mitgeliefert sind. Der Fairness halber sei aber erwähnt, daß bei einigen der Font-Editor DIDOT und das Grafikprogramm OUTLINE ART Pate gestanden haben.

Was bin ich?

Artworks Business ist in erster Linie ein Ideenlieferant. Typische Beispiele, so wie man sie im täglichen Bürobetrieb braucht, sind zusammengefaßt worden, um als Fundus für die Weiterentwicklung durch eigene Ideen zu dienen. Die Wege, wie man die Vorlagen entwickelt, sind verständlich und nachvollziehbar aufgezeigt. Gleichzeitig ist Artworks ein Gestaltungswerkzeug, das ebenso aufzeigt, wie die Beispiele leicht in eigene Entwicklungen umänderbar sind. Die Muster sind nicht irgendwann aus theoretischen Überlegungen entstanden, sondern waren zu etwa einem Drittel konkrete Aufträge einer Freiburger Werbeagentur, die diese vollständig in CALAMUS entwickelte. Exklusiv für die Verwendung im Artworks-Paket hat der Copyright-Inhaber auf das Urheberrecht verzichtet, so daß jeder Käufer über diese Gestaltungsmuster verfügen kann.

Schlußgedanken

Mit dem ersten Artworks-Paket „Business" ist uns eine Anwendungsrichtung aufgezeigt worden, die wir bisher nicht kannten. War es doch in der Vergangenheit so: Man kaufte sich ein Programm für DTP, wälzte das Handbuch, suchte nach Praxisbeispielen. Diverse Bücher zu diesem Anwendungsfall im allgemeinen und zu CALAMUS im besonderen waren durchweg ebenso theoretisch wie das Originalhandbuch. Selbst sogenannte „Praxisbücher" hatten nicht viel mehr als nur das sture Abspulen bestimmter Arbeitsschritte zu bieten. Da kann es nicht selten passieren, daß man ebenso schnell frustriert Programm und Bücher in die Mülltonne wünscht, weil bei dem geringsten Fehler alles in sich zusammenfällt.

Artworks hat sich zum Ziel gesetzt, das Pferd von hinten aufzuzäumen. Man legt uns eine Vielzahl fertiger Muster vor und zeigt uns, wie das erreicht wurde. Darüber hinaus sind aber auch die Mittel aufgezeigt, aus diesen Mustern selbst etwas zu gestalten. Artworks liefert uns sozusagen den fertigen Kuchen (zum Kosten) nebst Rezept (zum Lernen) und Zutaten (zum Ausprobieren) - ist dieser Weg nicht der bessere?

Das Handbuch ist leicht zu lesen und ausführlich. Die Strukturierung der Kapitel bildet eine logische Folge von Basistheorie bis zum fertigen Ausdruck. Auch an der Gestaltung des Seitenlayouts selbst sieht man, daß Profis am Werk waren. An verschiedenen Stellen hätte ich mir aber gewünscht, etwas mehr Bilder zu sehen. So sind bedauerlicherweise bei der Erklärung der Druckverfahren sowie zur Aufbereitung und Präsentation keinerlei Skizzen oder Grafik zu sehen. Auch sonst sind die Kapitel nicht gerade mit vielen Bildern bestückt. Hier wäre wohl wieder einer meiner Lieblingssprüche angebracht: „Ein Bild sagt mehr als tausend Worte".

Auch hätte man das Handbuch nicht unbedingt auf 120-Gramm-Karton drucken müssen, was es nur unnötig dick werden läßt. (Übrigens: Unser normales Schreibmaschinenpapier hat ein Gewicht von 80 Gramm pro Quadratmeter.)

Alles in allem bietet dieses Gestaltungspaket zu einem Preis von 398 DM einen reichen Schatz an Ideen. Ich glaube fest, daß uns, bedingt durch diese neue Art der Konfrontation mit einem Anwendungsprogramm, alsbald noch mehr Beispiele präsentiert werden (vielleicht für Glückwunschkarten, Schülerzeitungen, Werbebriefe oder Lehrbücher?). Der Anfang ist gemacht.

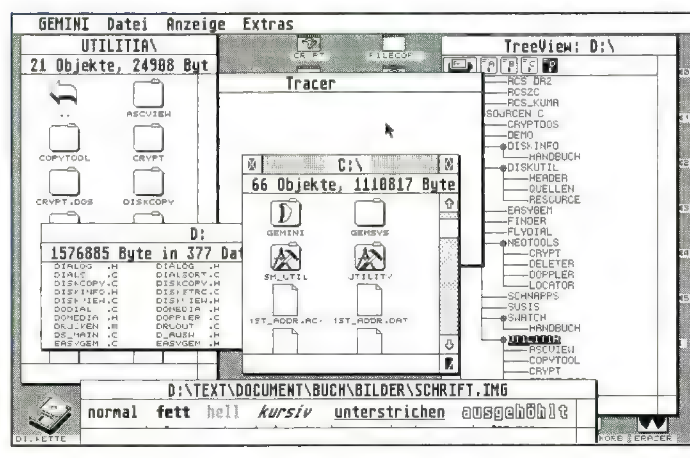
DK

Bezugsquelle:

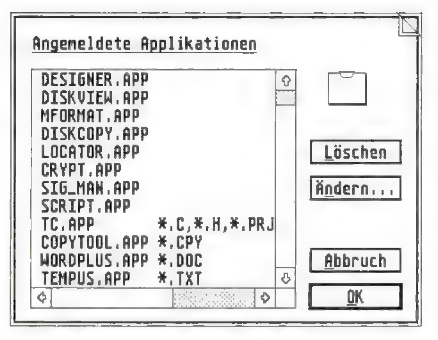
Hesse & Herwig GbR
Lesteweg 33
4790 Paderborn
Telefon: 05251 92231

Shareware-Desktop GEMINI

Version 1.2



In der Herde schwarzer Schafe im Public Domain- und Shareware-Bereich tummeln sich doch hin und wieder weiße Schafe. (Eigentlich sollte es ja anders herum sein, aber leider...) Eines dieser „weißen“ Schafe und damit eine rühmliche Ausnahme stellt das Shareware-Desktop GEMINI dar, das vor etwa einem Jahr erstmals die ST-Besitzer/innen erreichte. Wir wollen Ihnen nun die neueste Version (1.2) vorstellen, die seit November 1990 vorliegt.



GEMINI ist eine Shell, deren Autoren viel Wert auf saubere Programmierung gelegt haben. So sind beispielsweise die Richtlinien zur Gestaltung der Oberfläche eingehalten, womit die Bedienung relativ einfach gemacht ist. Daß ein Programm unter Beachtung bestehender Richtlinien nicht an Funktionalität verliert, sondern vielmehr gewinnt, beweist diese Shell.

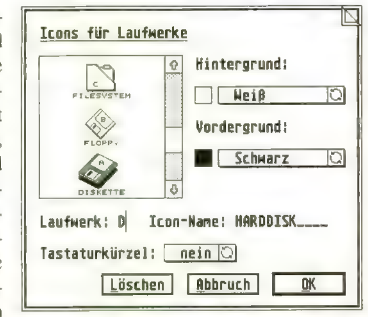
GEMINI ersetzt das Original-Desktop des Atari ST bzw. TT. Es kann ab TOS 1.04 (Rainbow-TOS; die Versionsbezeichnung war vormals 1.4) automatisch gestartet werden, so daß der ST bzw. der TT sofort mit der GEMINI-Oberfläche erscheint. Alle vom Desktop her bekannten Funktionen (mit einer kleinen Ausnahme) lassen sich auch unter GEMINI ausführen. Darüber hinaus bietet GEMINI sehr viele Funktionen, die unter dem Atari-Desktop schon immer vermißt wurden. Auch der TT-Desktop bietet nicht das, was GEMINI bietet.

Zuerst ein Funktionsüberblick. GEMINI beinhaltet zwei Programme: Mupfel und Venus. Mupfel ist eine kommandoorientierte Oberfläche, die UNIX-ähnlich gehalten wurde. Venus bietet mit der grafischen Oberfläche die gewohnte, einfache Bedienung mit der Maus. Über die grafische Oberfläche sind die Standardfunktionen wie Dateien, Ordner oder Disketten kopieren und löschen erreichbar. (Hier findet sich auch die oben angesprochene Ausnahme: GEMINI kann keine Disketten sektorenweise kopieren. Die Autoren verzichteten angesichts der vielen zu erwartenden Sonderwünsche auf diese Funktion.) Natürlich lassen sich die Namen von Dateien, Ordnern und Disketten ändern sowie Dateien und Ordner bewegen. GEMINI 1.2 kann nun im Gegensatz zur ersten Version auch Disketten formatieren.

Console-Fenster
T-Mail
T-Mail Bold
Courier
Rufus
TMAIL ANSI
9
18
Zeilenzahl: 30
Spaltenzahl: 80
Mupfel fc
Invertierter Text
invertiert
fett
unterstrichen
Abbruch
OK

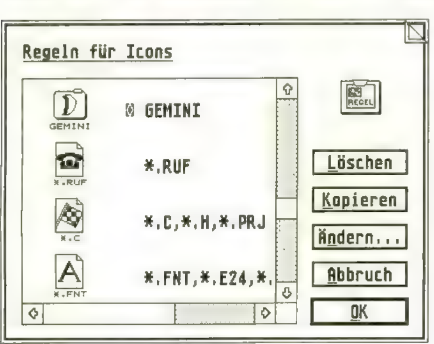
Die Inhaltsverzeichnisse von Ordnern und Disketten werden in Fenstern dargestellt. Die Darstellung erfolgt als Icon (in zwei Größen) oder als Text, wobei der Zeichensatz dank GDOS frei wählbar ist. Die Angabe der Größe, des Datums

und der Zeit ist optional. Für Dateien und Ordner können die Icons nach bestimmten Regeln festgelegt werden. Programme, C-Quelldateien und vieles mehr sind dadurch schnell zu erkennen. Ein besonderer Vorteil ist, daß alle Icons auf dem Desktop abgelegt werden können. Häufig benutzte Programme, Ordner und Dateien sind damit immer schnell greifbar. GEMINI 1.2 verfügt über eine Aufräumfunktion, mit der die Icons auch gut und schnell plazierbar sind. Die Probleme der ersten GEMINI-Version sind damit beseitigt. Programme können jetzt als Overlay gestartet werden. Es bleibt nur noch ein kleiner Kern im Speicher. Besitzer/innen kleinerer Rechner werden sich darüber freuen!

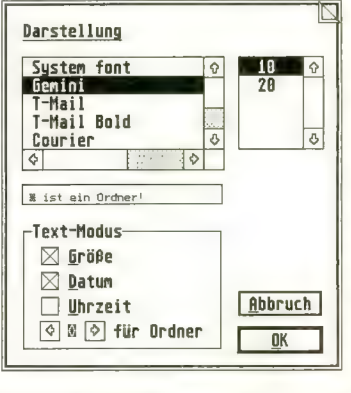


Die Fenster von GEMINI sind weitgehend über Tastatur bedienbar. Nett sind so kleine Hilfsmittel wie das Öffnen eines Ordnerfensters durch Doppelklick auf den Ordner (und den Verlust des Inhaltsverzeichnisses), wobei bei gleichzeitig gedrückter Alternate-Taste das Ordnerfenster zusätzlich zum vorigen geöffnet wird. Ferner kann GEMINI 1.2 jetzt auch Dateien anzeigen bzw. ausdrucken, wenn diese nicht für eine bestimmte, vordefinierte Anwendung gedacht sind. Programme können - wie gerade vom Atari-Desktop her bekannt - als Anwendung für Dateitypen angemeldet werden. Unter GEMINI lassen sich jedoch mehrere Dateitypen anmelden. Ein C-Entwicklungssystem kann so über Dateien mit den Extensions C und H gestartet werden.

Die Kommandooberfläche enthält einen reichhaltigen Satz von Befehlen. Für diverse Anwendungen lassen sich praktische Batch-Dateien erstellen. Auch Befehle zur Anzeige der Cookie-Variablen oder zum Ändern der ab TOS 1.04 relevanten Bits in Programm-Headern sind dabei! Der volle Befehlssatz ist in aller Kürze in unserer Liste aufgeführt. Ausgaben von TOS-Programmen, die sonst auf dem vollen Bildschirm angezeigt werden, lenkt GEMINI in das Kommandofenster um. Die gewohnte GEM-Oberfläche muß also nie verlassen werden!



GEMINI enthält noch viele kleine Neuigkeiten, die hier (ohne Anspruch auf Vollständigkeit) aufgeführt werden sollen. Datei-Icons lassen sich in das Kommandofenster ziehen, womit die Dateinamen automatisch übertragen werden. Teile aus einer Kommandozeile lassen sich ins Clipboard kopieren und bei Bedarf wieder einsetzen. Bei geöffnetem Kommandofenster kann weitergearbeitet werden. Ein Doppelklick auf einen Dateinamen bei gehaltener Control-Taste bewirkt die Ausgabe einer zur Datei gehörigen Information. Neue Icons für die grafische Oberfläche. Komfortable Änderungsmöglichkeiten für Icon-Regeln (mit Kopieren!). Shortcuts für bis zu 10 Dateien, Ordner oder Laufwerke.



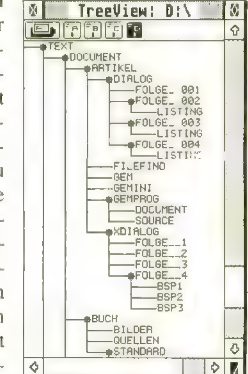
Soweit der Kurzüberblick über GEMINI 1.2. Da GEMINI nach dem Shareware-Konzept vertrieben wird, haben alle Anwender/innen den Vorteil, das Programm vor Entrichtung des Shareware-Obulus' ausreichend zu testen. Sie können sich also selbst eingehend von den Vorteilen dieser Shell überzeugen. Das Shareware-Konzept bedeutet, daß das Programm ohne Erhebung von Kopiergebühren beliebig weitergegeben werden darf. Bei einer Nutzung ist der Shareware-Obulus (bei GEMINI 50.- DM) zu entrichten. Das Shareware-Konzept unterscheidet sich damit grundlegend von Public Domain oder Freeware, da die Entrichtung des Obulus' bei einer Nutzung Pflicht ist. Leider halten sich längst nicht alle Anwender/innen von Shareware-Programmen an diese Regel. Genutzte Shareware-Programme kommen Raubkopien gleich, sofern der Test beendet und der Obulus nicht entrichtet wurde. In Verbindung mit GEMINI existieren diverse Zusatzprogramme, die die interne, dokumentierte Schnittstelle mehr oder weniger gut ausnutzen. An dieser Stelle sollen zwei Accessories genannt werden. Zum einen ist dies das Accessory CLIPBRD, das das Clipboard verwaltet, und zum anderen TreeView, das die Directory-Hierarchie im Fenster darstellt.

Das CLIPBRD-Accessory beinhaltet eine Funktion zur Anzeige von Dateien in Standardformaten. Legt man beispielsweise das Accessory-Icon von CLIPBRD auf dem GEMINI-Desktop ab, kann eine Bilddatei im Bit-Image-Format (Extension IMG) darauf geschoben werden. CLIPBRD zeigt das Bild umgehend in einem Fenster an. Die simple *Datei anzeigen*-Funktion ist so auch für Bilder erweitert!

TreeView zeigt die komplette Directory-Hierarchie (auch ab einer bestimmten Ebene) an. Ordner können durch Klick auf Knotenpunkte aus- und wieder eingeblendet werden. Klickt man auf einen Ordnernamen, öffnet GEMINI automatisch das dazugehörige Fenster, womit auch die zum Ordner gehörigen Dateinamen sichtbar werden! Sogar in Funktionen wie dem zyklischen Wechseln des aktiven Fensters ist das TreeView-Fenster mit einbezogen!

Neben diesen beiden Utilities existieren noch weitere (wie DISKINFO 3.00), die das GEMINI-Protokoll zumindest soweit beherrschen, daß sie sich über Icons auf dem Desktop starten lassen.

Ein Jahr GEMINI-Erfahrung bedeutet auf der einen Seite, eine Shell schätzen zu lernen, aber auch, sich über schlechte Programme (vor allem im PD-Bereich) zu ärgern. So zum Beispiel birgt der Vorteil von GEMINI, einen beliebigen GEM-Font für das Kommando- und die Directory-Fenster zu verwenden, die Gefahr, mit Programmen zu kollidieren, die ihre virtuelle Workstation nicht sauber öffnen und somit nicht mit dem GDOS zu-



DAS VORBILD



Unser System setzt Maßstäbe

in Leistung, Zuverlässigkeit und Geschwindigkeit, ist kaum zu hören und natürlich vollkommen im Atari ST Design gehalten. Die ausgefeilte, grafische Benutzerführung macht die Bedienung spielend leicht.
... mit weniger Qualität sollten Sie sich nicht zufrieden geben.

Lieferbare Ausführungen

SCSI Ultra Speed Drive

Ausführung	Preis
52 MB · 17 ms · 64 KB Cache	1498,-
85 MB · 19 ms · 64 KB Cache	1998,-
105 MB · 17 ms · 64 KB Cache	2198,-
120 MB · 15 ms · 64 KB Cache	2498,-
170 MB · 15 ms · 64 KB Cache	2998,-
210 MB · 15 ms · 64 KB Cache	3498,-

SCSI Speed Drive

Ausführung	Preis
49 MB · 28 ms	1398,-
85 MB · 24 ms	1648,-
Wechselplatte SCSI Speed Drive 44	1998,-
Streamer SCSI Speed Drive 155	2298,-

Festplatten am Atari TT

Durch ACSI/SCSI Umschaltung und SCSI Tools 2.0 Betrieb am SCSI Port des TT möglich.

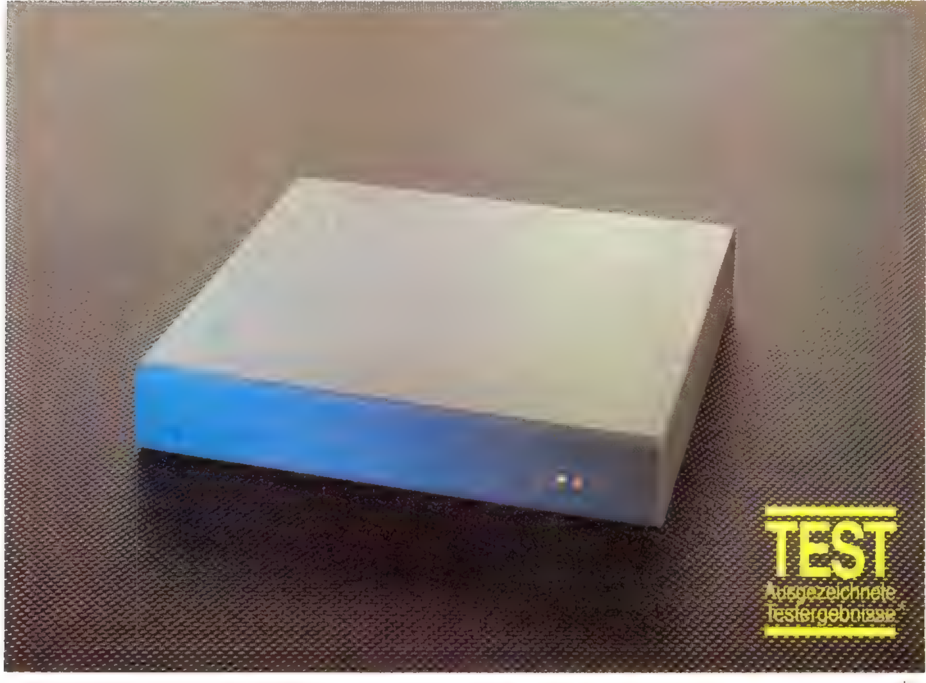


intern voller SCSI Standard



SCSI Schnittstelle an Geräterückseite herausgeführt

SCSI Ultra Speed Drive



* [illegible] Preis/Leistungsverhältnis« (ST Computer 4/90)
[illegible] auf dem Festplattenmarkt« (ST Magazin 8/90)

Speichererweiterungen

Unsere Speichererweiterungen entsprechen dem neuesten Stand der Technik. Ingenieurmäßiges Schaltungsdesign, die Fertigung großer Stückzahlen auf hochmodernen Industriestraßen sowie ausgefeilte Maßnahmen zur Qualitätssicherung setzen einen hohen Qualitätsstandard, von dem auch Sie profitieren können.

Unsere Speichererweiterungen laufen in allen Rechnern. Durch gezielte Maßnahmen konnte die Stromaufnahme und die Störanfälligkeit bei knappem Bustiming deutlich reduziert werden. Dadurch können unsere Erweiterungen auch in Rechnern mit IMP-MMU (bei IMP MMU nur gleich große Bänke möglich, oder in Atari 1040 ST problemlos betrieben werden.

Völlig neue Ideen bei Speichererweiterungen wurden bei unserer Multi-MB-Erweiterung gegangen. Die lange Erfahrung in Konstruktion und Entwicklung sowie das Bemühen um größtmögliche Flexibilität sind in die Entwicklung dieses Produktes eingegangen. Und das hat sich gelohnt.

Mit ein und derselben Karte können Sie zum Beispiel einen
– Atari 1040 ST von 1 MB auf 2 MB
– einen Atari 260/520 von 512 KB auf 1 MB,
– einen Atari 260/520 von 512 KB oder 1 MB auf 2 MB,
– einen Atari Mega ST 1 von 1 MB auf 2 MB
– und
aufrüsten, egal, ob in Ihrem Rechner eine IMP-MMU sitzt oder nicht.

NEUHEIT Multi-MB-Board

Sie wollen Ihren Rechner auf 4 MB erweitern? Kein Problem. Hierzu stecken Sie einfach eine Zusatzplatine Huckepack auf das Multi-MB-Board und schon haben Sie 4 MB. Einfacher und universeller geht es nicht.

Und dabei ist unsere Multi-MB-Erweiterung auch noch sparsam. Es werden immer nur soviele RAMs gesteckt, wie auch wirklich benötigt werden. Wurde früher eine voll mit 2 MB bestückte Platine benötigt, um einen 520 ST oder 1040 ST aufzurüsten, so benötigen Sie mit dem Multi-MB-Board nur 1 MB bzw. 1,5 MB um den Speicher aufzurüsten. Und wenn wir RAMs sparen – dann sparen Sie Geld.

Unser Angebot umfaßt eine große Palette an professionellen Speichererweiterungen, die generell in zwei Versionen lieferbar sind.

Die vollsteckbare Version ist kinderleicht einzubauen. Sämtliche Verbindungen der Speicherkarte können ohne Lötarbeiten vorgenommen werden. Voraussetzung für den Einbau dieser Version ist, daß im Rechner die MMU sowie der Videoshifter gesockelt sind. Der Steckverbinder zur MMU besitzt natürlich vergoldete Kontakte. Die teilsteckbare Version ist für Rechner mit nicht gesockelten Bauteilen (Shifter und/oder MMU) und für alle, denen das Anlöten von ca. 18 Lötverbindungen keine Probleme bereitet, gedacht.

Zum Lieferumfang jeder Speichererweiterung (bestückt) gehört eine ausführliche und bebilderte Einbauanleitung sowie ein Speichertestprogramm. Selbstverständlich wird jede unserer Speichererweiterungen vor dem Versand im Rechner stückgeprüft.

Sollte dennoch eine Frage offen bleiben, so helfen Ihnen an unserer Service-Hotline versierte Techniker gerne weiter.

Modell 1 S: Speicheraufrüstung auf 1 MB, voll steckbar, 198,00 DM

Modell 2: Speicheraufrüstung auf 2,5 MB, teilsteckbar (beim Mega ST 2 Aufrüstung auf 4 MB möglich) 398,00 DM, dto. Leerk. 189,00 DM

Modell 2 S: wie Modell 2, aber voll steckbar, mit vergoldeten Mikrokontakten, 449,00 DM, dto. Leerkarte 229,00 DM

Modell 2/4: Speicheraufrüstung auf 2,5/4 MB, teilsteckbar, Speicher in zwei Stufen 2,5/4 MB aufrüstbar. Auch für Mega ST geeignet. 449,00 DM, dto. als Leerkarte 249,00 DM

Modell 2/4 S: wie Modell 2/4, aber voll steckbar, mit vergoldeten Mikrokontakten. 498,00 DM, dto. als Leerkarte 289,00 DM

Multi-MB-Board, teilsteckbar, bestückt zur Aufrüstung von 1 MB auf 2 MB, 349,00 DM, nicht als Leerkarte erhältlich

Multi-MB-Board, vollsteckbar, bestückt zur Aufrüstung von 1 MB auf 2 MB, 398,00 DM, nicht als Leerkarte erhältlich

Speichererweiterung für 1040 STE, SIM-Module auf 2,5 MB 338,00 DM, auf 4 MB 676,00 DM

Diskettenlaufwerke

Diskettenlaufwerke: 3,5-Zoll- und 5,25-Zoll-Disketten-Laufwerke in vollendeter Qualität. Es werden nur die besten Materialien verwendet. Laufwerksgehäuse mit kratzfester Speziallackierung. 5,25-Zoll-Laufwerk (720 KB/1,2 MB) incl. beige Frontblende, 40/80 Track Umschalter, Software IBM-Atari anschlußfertig 289,- DM, Chassis Atari modifiziert 198,00 DM, 1,44-MB-Laufwerk incl. HD-Interface, anschlußfertig 298,- DM, 3,5-Zoll-LW incl. beige Frontblende mit NEC FD 1037 oder TEAC FD 235 anschlußfertig 239,00 DM, Chassis 169,00 DM

Atari-Zubehör

Auto-Monitor-Switchbox: A.R.S. (Automatic Resolution Selection) Das Programm wird automatisch in der richtigen Auflösung gestartet (nur TOS 1.0 und 1.2). Mit der Auto-Monitor-Switchbox können Sie über die Tastatur zwischen Monochrom und Farbmonitor umschalten oder einen Tastaturreset durchführen. Die mitgelieferte Software ist resetfest. Durch Einbinden der von uns mitgelieferten Routinen Umschaltmöglichkeit ohne RESET. Zusätzlicher BAS und Audio Ausgang. Auto-Monitor Switchbox 59,90 DM, Auto-Monitor Switchbox Multisync 69,90 DM, weitere Modelle von 29,90 DM bis 69,90 DM

Video Interface +: ermöglicht die Farbwiedergabe an einem Farbfernseher, Monitor oder Videorecorder mit Videoausgang (mit integrierter Auto-Monitor-Switchbox-Funktion) 159,00 DM

Neu: Echtzeit-Videodigitalisierer in 16 Graustufen (Einlesen von Videosignalen im Computer, kein Standbild erforderlich) 398,00 DM

HF-Modulator: zum Anschluß des Atari ST an jeden gewöhnlichen Farbfernseher. Der Ton wird über den Fernseher übertragen. 189,00 DM, Aufpreis Monitorswitchbox 30,00 DM

Festplattenzubehör, wie SCSI-Hostadapter, Einschaltverzögerungen, 1,2 m- DMA-Kabel etc.

STTAST II: ermöglicht den Anschluß einer beliebigen PC-(XT-)Tastatur am ST, umschaltbare Mehrfachbelegung der Tastaturbelegungen, freie Programmierbarkeit von Makros und Generieren von Start-Up-Files (mit AUTO Load), Tastaturreset, unterstützt auch PC Ditto und PC/AT Speed 149,00 DM. Set: PC Tastatur mit Mikroschalter + ST Tast II 298,00 DM

Abgesetzte Tastatur am ST: Tastaturgehäuse mit Spiralkabel, Treiberstufe, Resettaste und Joystickbuchsen eingebaut. Computertyp angeben. 109,00 DM

Towergehäuse: nur Gehäuse oder mit kundenspezifischer Bestückung ab 298,00 DM

RTS Tastaturkappen: ab 89,00 DM

Uhrmodul intern: die Bootsoftware befindet sich auf ROM's im Betriebssystem. Wichtig: Betriebssystem angeben: ROM TOS oder Blitter TOS. 119,00 DM

Floppyswitchbox: ermöglicht den Anschluß von drei Laufwerken am ST. Ausgestattet mit speziellen Treibern für 3,5 und 5,25 Laufwerke. Computertyp angeben. 89,00 DM

Außerdem: HD-Kid – interne Beschaltung incl. Software zum Anschluß von HD-(3,5"- und 5,25"-)Laufwerken am ST. Keine Zusatzschaltung im Laufwerk erforderlich. 69,00 DM

Gehäuse – 3,5" und 5,25"-Gehäuse incl. Montagematerial 29,90 DM, Festplattengehäuse (Mega-Format) 99,00 DM

SCSI-Hostadapter – orig. ICD Hostadapter Advantage+ (mit Uhr) 298,00 DM, Advantage (ohne Uhr) 269,00 DM, Mikro 249,00 DM. Lieferung incl. DMA-Kabel, ICD Software und Dokumentation

Einschaltverzögerung – Ermöglicht das gleichzeitige Einschalten von Computer und Festplatte. Einbau erfolgt im Computer durch einfaches Aufstecken. Zeitverzögerung einstellbar, 49,90 DM

Tastaturkabel Mega ST – langes Tastaturkabel (2 m) für den Mega ST 29,90 DM

Eprombrenner – orig. MAXON Junior-Prommer, 229,00 DM, Epromkarte 128 KB incl. Gehäuse (Steckmodul), 49,90 DM

Laserinterface II – Abschalten des Laserdrucker im Festplattenbetrieb möglich. Lieferung komplett mit Netzteil VDE GS, 79,00 DM

HD-Kid – Einbau ohne Auslöten des Floppycontrollers möglich 69,00 DM

MS-DOS-Emulatoren – MS-DOS-Emulator zum internen Einbau ATonce 479,00 DM, AT Speed 479,00 DM. Super-Charger MS-DOS-Emulator zum externen Anschluß 729,00 DM

Hypercache – 16-MHz-Erweiterung mit 16 KB Cache (orig. PROVME), 550,00 DM. Auch gegen Aufpreis als besonders stromsparende CMOS-Version lieferbar

Fordern Sie unseren Gesamtkatalog an.

Speichererweiterungen
Festplatten
Wechselplatten
Streamer
Diskettenlaufwerke
HD-Diskettenlaufwerke
HD-Kids
Laufwerkschassis
Monitorumschaltboxen
Videointerface
Videodigitalizierer
HF-Modulator
SCSI-Hostadapter
Einschaltverzögerungen
PC-Tastatur an Atari ST
Towersysteme
Towerzubehör
abgesetzte Tastatur am ST
RTS-Tastaturkappen
Echtzeituhren
Floppyswitchbox
Laufwerksgehäuse
Software (PD)
Disketten
Festplattengehäuse
Computerkabel
langes Mega-Tastatur-kabel
Eprombrenner
Epromkarten
Laserinterface II
ATonce/AT Speed
Supercharger
Hypercache

Hard & Soft A. Herberg

Obere Münsterstr. 33-35 · 4620 C.-R. · ☎ (0 23 05) 1 80 14 · Fax 3 24 63

exit:	Mupfel (Kommando-Oberfläche) beenden
cd:	Directory wechseln
pwd:	aktuelles Directory anzeigen
x:	Laufwerk wechseln (beispielsweise C: oder M:)
pushd:	angegebenes Directory merken
popd:	zuletzt gemerktes Directory holen
dirs:	alle gemerkten Directories anzeigen
help:	Hilfe zu allen internen Kommandos
cp:	Dateien kopieren
mv:	Dateien umbenennen
rename:	Dateien umbenennen, wobei die Extension geändert wird
rm:	Dateien löschen
mkdir:	Directory anlegen
rmdir:	Directory löschen
chmod:	Dateiattribute ändern
touch:	Änderungsdatum von Dateien aktualisieren
ls:	Directory-Inhalte anzeigen
echo:	Ausgabe der übergebenen Argumente
cat:	Dateiinhalt anzeigen
more:	Dateiinhalt seitenweise anzeigen
print:	Dateien ausdrucken
whereis:	Kommando suchen
df:	Speicherplatz von Massenspeichern anzeigen
label:	Label von Massenspeichern anzeigen (und ändern)
backup:	Dateien kopieren, wobei die Extension au BAK geändert wird
tree:	Directory-Hierarchie anzeigen
find:	Dateien suchen
hash:	Hash-Tabelle für externe Kommando anzeigen
setenv:	Environment anzeigen bzw. ändern
show, env:	Environment anzeigen
set:	Variablen anzeigen oder ändern
var:	Variablen setzen oder ändern
alias:	Alias-Namen anzeigen oder ändern
noalias:	Kommando ohne Aliasing ausführen
fkey:	Funktionstastenbelegung anzeigen oder ändern
date:	Datum anzeigen oder ändern
wc:	Zeilen, Worte und Zeichen in Dateien zählen
du:	Platzbedarf eines Directorys ermitteln
file:	Dateityp ermitteln
init:	Disketten initialisieren (löschen)
format:	Disketten formatieren
timer:	Timer setzen
setscrap:	Scrap-Pfad des AES setzen
blitmode:	Blitter-Modus anzeigen oder ändern
version:	Versionsnummern anzeigen
shrink:	Speicher verkleinern
free:	freien RAM-Speicher anzeigen
pause:	Tastendruck abwarten
cookie:	Inhalt des Cookie-Jars anzeigen
kbrate:	Tastatur einstellen
kclick:	Tastenklick einstellen
rsconf:	Baud-Rate und Übertragungsprotokoll einstellen
prtconf:	Druckerparameter einstellen

Tab.1: Die Befehle der GEMINI-Shell

rechtkommen. Oder TOS-Programme, die direkt den Bildschirm beschreiben und somit die I/O-Redirection, die für die Ausgabe des Textes im Kommandofenster zuständig ist, zunichte machen. Oder die zahlreichen sogenannten GEM-Programme, die mit immer neuen Oberflächen kommen, nur weil ihre Autorinnen und Autoren der Meinung sind, die beste Oberfläche gefunden zu haben. Eine Umgewöhnung von der GEM-Oberfläche, derer sich GEMINI bedient, ist viel zu häufig nötig. (Um es an dieser Stelle noch einmal deutlich auszudrücken: Es liegt nicht an GEMINI, wenn andere Programme unsauber programmiert sind!)

Es gibt jedoch einen Zeitpunkt, zu dem sich ein stabiles System eingespielt hat. Bei fehlerhaften Programmen kann man sich bei den zuständigen Software-Entwicklern und -Entwicklerinnen beschweren und auf eine Fehlerbehebung drängen.

Liegt erst ein funktionstüchtiges System vor, möchte man die Vorteile von GEMINI nicht mehr missen. Für ein Programmierprojekt werden die wichtigsten Dateien auf dem Desktop abgelegt und je nach Bedarf benutzt. Wichtige Ordner und Programme sind schnell und ohne große Suchereien in Directory-Hierarchien greifbar. Für sich häufig wiederholende Befehlsfolgen lassen sich praktische Batch-Dateien anlegen.

Dietmar Rabich

Bezugsadressen:

GEMINI (mit AMCGDOS 4.0, TreeView, Chameleon und einigen weiteren kleinen Utilities) erhalten Sie bei Gereon Steffens, Elsterweg 8, 5000 Köln 90, oder bei Stefan Eissing, Dorfbauerschaft 7, 4419 Laer-Holthausen, gegen eine formatierte Diskette und einen Freiumschlag. Die Shareware-Gebühr beträgt 50,- DM.

TreeView ist von Stephan Gerle, Ruthstr. 8, 4600 Dortmund 1.

CLIPBRD ist Public Domain (ST-Computer-PD 320).

Interface für Laserdrucker



Es war schon eine kleine Sensation, als die Firma Atari seinerzeit einen extrem preisgünstigen Laserdrucker herausbrachte. Endlich war eine repräsentative und hochwertige Druckqualität zu einem revolutionären Preis erschwinglich geworden. Besonders in puncto Druckgeschwindigkeit machte der Atari-Laser allen seinen großen Vorbildern aus der MS-DOS-Szene etwas vor: Er brachte es auf 8 Seiten pro Minute.

Die Vorzüge dieses Druckers mußten mit einigen Restriktionen erkauft werden: kein interner Arbeitsspeicher, folglich Mitbenutzung des Speichers in der Zentraleinheit (wohl dem, der einen MEGA ST 4 hat), keine Anschlußkompatibilität zu anderen Computern. Was als extrem störend empfunden wurde, war einerseits der Umstand, daß der Drucker immer eingeschaltet sein mußte, um keine Systemabstürze im Rechner zu verursachen. Ganz dramatisch ist das Katz-und-Maus-Spiel, das der DMA-Bus mit einer zusätzlich angeschlossenen Festplatte vollführt, wenn ein angeschlossener Laserdrucker keinen Betriebsstrom hat. Andererseits vollführte der Lüfter unseres Druckers mit lautem Surren ständig „akustische Umweltverschmutzung", was auch nicht jedermanns Sache war. Dann gab es ganz findige Zeitgenossen, die das Problem mit dem „Klappentrick" lösten: Einfach die Papierauswurfklappe nach hinten öffnen - und schon sorgt ein kleines Schalterchen dafür, daß der Drucker in den „Nicht-Bereit-Status" verfällt. Das Lüftergebläse schaltet ab, und es kehrt Ruhe ein. Andere Tüftler haben einfach in den Stromkreislauf dieses Schalterchens einen weiteren Kippschalter eingebaut und täuschen damit das Öffnen der Auswurfklappe vor, der Effekt ist derselbe.

In letzter Zeit waren Warnungen zu vernehmen, die Auswurfklappe nicht unbedingt für längere Zeit offenstehen zu lassen, weil sich Staubteilchen dann vermehrt in die Mechanik „verirren" würden. Alles in allem war das gesamte Handling mit dem Laserdrucker nicht gerade befriedigend, und leider hat sich trotz einiger Beschwerden in der Firma Atari zu diesem Problem überhaupt nichts getan. Wieder einmal war es nötig, daß sich andere Köpfe über die Lösung solcher Probleme hermachten.

Schalldämpfer für den Laserdrucker

Wie wir bereits erkennen mußten, ist es nicht möglich, den Laserdrucker einfach ausgeschaltet zu lassen; und ständig den DMA-Port-Stecker lösen ist auch nicht zu empfehlen. Der Grund liegt im prinzipiellen Aufbau des Gerätes und seinem Anschluß am DMA-Port. Der Atari-Laserdrucker besteht bekannterweise aus zwei Geräten, dem eigentlichen Laserdrucker (Name: SLM 804) und dem DMA-Controller (Name: SLMC 804), welcher zwischen Laser und Computer geschaltet wird. Der DMA-Controller dient zur Umsetzung des 8-Bit-DMA-Busses auf die sogenannte Serial-Video-Line des Laserdruckers und zur Ansteuerung desselben. Hier werden die DMA-Adresse decodiert und die Atari DMA-Laserbefehle von einem Ein-Chip Microcontroller (6801) in für das Druckwerk verständliche Form gebracht. Zusätzlich wird der DMA-Bus für nachfolgende Geräte (vielleicht eine Festplatte) durchgeschleift.

Der Bus findet hierbei seine Ansteuerung über einen Transceiver-Chip (SN74-LS245), der den Bus bidirektional (also in zwei Richtungen) bedienen kann. Leider wird derkomplette DMA-Controller über den Druckeranschluß des Laserdruckers mit Strom versorgt. Dies hat zur Folge, daß beim Abschalten des Laserdruckers der DMA-Bus abstürzt, da auch der DMA-Controller ohne Stromversorgung ist.

Dieses Problem ist leider vollkommen unabhängig von der Anschlußweise des DMA-Controllers. Wenn Sie den Controller wie im Laser-Handbuch beschrieben als erstes Gerät mit dem Computer verbinden und anschließend eine Festplatte direkt an den durchgeschleiften DMA-Bus des Controllers anschließen, blockiert der Transceiver-Chip sowieso den DMA-Bus (bei abgeschaltetem Laserdrucker). Das ist leider auch so, wenn Sie den Laserdrucker als Endgerät (also erst die Festplatte) anschließen. Eine Nichtbeachtung dieser Tatsachen kann zu irreparablem Datenverlust auf der Festplatte führen.

Seit einiger Zeit gibt es für dieses Problem unter anderem das „Laserinterface SLM 804" der Firma digital image, welches dieses Problem durch einfaches Aufstecken zwischen DMA-Controller und dem Laserdrucker löst. Dem Computer bzw. der Software wird bei abgeschaltetem Laserdrucker ein noch nicht fertig aufgeheizter signalisiert, so daß bei einem

versehentlichen Druckbefehl die Meldung *WARM-UP* (d.h. Drucker noch nicht fertig aufgeheizt) erscheint. Man kann dann im laufenden Programm, ohne Warm- oder Kaltstart, einfach den Laserdrucker anschalten und ausdrucken. Der Controller und der Laserdrucker werden vom Laserinterface initalisiert und können beliebig an- und abgeschaltet werden.

Fazit: Wieder einmal mußte eine unabhängige Firma Lösungen für Unzulänglichkeiten anbieten, wozu Atari als Weltunternehmen nicht imstande war. Wer das lästige Surren des Laserdruckers kennt, der wird heilfroh sein, daß es endlich eine befriedigende Lösung gibt. Der Preis von 98 DM darf als Beitrag zur Verbesserung der „Umweltakustik" gewertet werden, und im Büro kann man endlich wieder mit geringerer Phonzahl Unterhaltungen führen. Mein Urteil: ein sinnvolles Zubehör.

DK

Bezugsquelle:
digital image EDV-Service
Hocheimer Straße 21
6095 Gustavsburg
Telefon: 06134 / 51706

Trenne nie ST...?

Silbentrennung für Wordplus und Script

ScripTRENN Silbentrennung

© Oliver Völckers / APiSoft
Bundesallee 56, 1 Berlin 31
123 Anschläge, 20 Trennungen

√Trennen √Sichern

1st TRENN Silbentrennung

von Oliver Völckers, Berlin
© Maxon GmbH, 6236 Eschborn
54 Anschläge, 14 Trennungen

OK √Trennen √Sichern

Bislang war Silbentrennung in vielen Textprogrammen eine umständliche Angelegenheit. Jetzt können Zusatzprogramme in 1st Wordplus und Script diese Aufgabe erledigen.

Eine Silbentrennung hat den Sinn, den Text gleichmäßig auf die Zeilen zu verteilen. Ohne sie würde bei linksbündiger Formatierung ein sehr ungleichmäßiger Rand entstehen: bei Blocksatz entständen Löcher. Je schmaler die Zeilen und je länger die Wörter sind, desto wichtiger ist ein gutes Trennverfahren. Da in der deutschen Sprache recht lange Wörter verbreitet sind, spielt die Silbentrennung bei der Textverarbeitung eine wichtige Rolle. Andererseits sind die Dudenregeln recht kompliziert und lassen sich nur schwer in Computerprogrammen ausdrücken.

Wordplus

Die frühen Versionen von 1st Word (damals noch ohne „Plus") kannten gar keine Silbentrennung. Erst Wordplus brachte die Möglichkeit, beim nachträglichen Umformatieren des Textes die Trennstellen anzugeben. Das Textprogramm macht Trennvorschläge, die mit einer Dialogbox bestätigt, verändert oder abgelehnt werden können. Weil die Vorschläge nicht besonders genau sind, ist das eine umständliche und zeitraubende Tätigkeit. Dennoch ziehen viele BenutzerInnen diese übersichtliche Dialogbox anderen Methoden vor.

Etwa jeder dritte Trennvorschlag von Wordplus ist falsch. Angesichts dieser Ungenauigkeit haben viele gemutmaßt, daß Wordplus gar nicht deutsche Trennregeln verwendet, sondern immer das englische Verfahren benutzt. Zumindest die Version 3.15 enthält jedoch einen Trennkatalog mit deutschen Silben. Auf die Zuverlässigkeit wirkt sich das leider nicht aus: Sie liegt eher noch niedriger als bei den älteren Versionen 1.89 und 2.02.; außerdem ist die Bildschirmdarstellung der Trennvorschläge häufig fehlerhaft.

Das Accessory 1st Trenn kann diesen Mangel ausgleichen. Nach dem Einschalten des Rechners bleibt es im Hintergrund und wartet darauf, daß Wordplus gestartet wird. Innerhalb von Wordplus ersetzt es die vorhandenen Trennregeln durch neue und sichere. Das Verfahren von 1st Trenn schafft eine Trefferquote von über 98%, das bedeutet ungefähr einen Fehler auf acht DIN A4-Seiten und ist ein deutlicher Fortschritt.

Jedesmal wenn der Zeilenumbruch mit der Funktionstaste F10 oder dem entsprechenden Menüeintrag ausgelöst wird, tritt 1st Trenn in Aktion und bedient die Trenn-Dialogbox automatisch. Das läßt sich am Bildschirm mitverfolgen; es sieht ganz lustig aus, wie der Trennstrich automatisch hin- und herrutscht, um an die richtige Position zu gelangen.

1st Trenn kann wahlweise vollautomatisch oder mit Bestätigung arbeiten. Bei eingeschalteter Bestätigung erscheint die übliche Wordplus-Dialogbox mit dem Titel *1st Trenn* und sinnvollen Vorschlägen. Da nur jeder fünfzigste Vorschlag fehlerhaft ist, wird dieses Spiel bald langweilig, und die meisten schalten auf Vollautomatik um. Die Geschwindigkeit der Silbentrennung ist hoch genug, um kaum ins Gewicht zu fallen: Ein neuer Randausgleich für eine zig-seitige Diplomarbeit z.B. dauert ohne Silbentrennung ca. drei Minuten, mit Silbentrennung geringfügig länger.

Zusätzlich verfügt 1st Trenn über eine Autosave-Funktion, die allerdings erst ab Wordplus 3.15 wirkt. Nach einer bestimmten Anzahl Tasteneingaben wird der bearbeitete Text automatisch gespeichert. Wer noch wenig Übung mit Computern hat, dem kann diese Funktion einige Sicherheit vor Datenverlust geben. 1st Trenn erkennt, ob eine Festplatte angeschlossen ist und speichert dann nach jedem Absatz. Bei Diskettenlaufwerken wird die Speicherung ungefähr nach jeder halben Seite ausgelöst.

Script

Das Textprogramm Script von Application Systems Heidelberg formatiert Texte gleich bei der Eingabe. Anders als bei Wordplus ist es nicht nötig, den Zeilenumbruch manuell auszulösen. Eine Trennhilfe entsprechend der Wordplus-Dialogbox wäre daher sehr lästig. Stattdessen erlaubt es Script, unsichtbare Trennhilfszeichen einzugeben, die dann beim Zeilenumbruch benutzt werden. Ginge das mit der Maus, wäre das eine bequeme Sache. Tatsächlich muß der Cursor aber an die richtige Stelle gebracht und dann Control+Bindestrich gedrückt werden. Dieser Wechsel zwischen Tastatur und Maus macht die meiste Mühe beim Silbentrennen mit Script.

Auch für Script gibt's ein Zusatzprogramm: Ähnlich wie 1st Trenn ist ScripTrenn ein Accessory von 30 kB Länge und nur innerhalb des Textprogramms aktiv. Auch hier ist die Silbentrennung 98%ig sicher, ansonsten ist die Arbeitsweise grundverschieden: ScripTrenn arbeitet bereits während des Tippens und gibt zwischen den Buchstaben blitzschnell das Trennhilfszeichen ein. Das hört sich sehr merkwürdig an, funktioniert jedoch erstaunlicherweise. Außer einem gelegentlichen Zucken des Cursors ist bei der Texteingabe keine Veränderung zu bemerken. Der Grund liegt in der hohen Geschwindigkeit von Script, die noch genug Reserven für ein solches Accessory hat.

Bei jedem Randausgleich und beim nachträglichen Verstellen der Absatzbreite berücksichtigt Script die von ScripTrenn eingegebenen Trennstriche. Dieser Randausgleich vollzieht sich außerordentlich schnell: zwanzig Seiten in sechs Sekunden! Beim Kopieren oder Verschieben von Textblöcken oder beim Suchen und Ersetzen bereiten die Trennstriche keine Probleme. Alternativ zur automatischen Trennung lassen sich die Trennhilfszeichen manuell mit der Maus setzen; das ist immerhin bequemer als über die Tastatur. Zusätzlich bietet ScripTrenn die gleiche Autosave-Funktion wie 1st Trenn.

Da ScripTrenn während der Eingabe arbeitet, kann es von Diskette geladene Texte nicht nachträglich mit Trennstrichen versehen. Diese Aufgabe erledigt ein im Lieferumfang enthaltenes Accessory mit dem sinnigen Namen ScripTipp. Es lädt ASCII oder Wordplus-Dateien und „tippt" sie mit Trennstrichen versehen in Script ein. Das geht nicht direkt mit Script-Textdateien, diese müssen zunächst im ASCII-Format gespeichert und dann erneut eingelesen werden; ScripTipp ist also bloß ein Behelf. Da die vorhandenen Script-Dateien in der Regel fertig formatiert und ggf. von Hand getrennt vorliegen, ist dieser Mangel nicht so schwerwiegend.

Sowohl 1st Trenn als auch ScripTrenn laufen zuverlässig auf allen Atari STs und sogar auf dem TT. Die Bildschirmauflösung ist egal, und mit anderen Programmen oder Accessories haben sich bislang keine Konflikte ergeben. Ausnahme: Bei Shells, die das Textprogramm indirekt starten, wird die Silbentrennung nicht aktiviert. ScripTrenn funktioniert mit Script 1.0 von 1989 und mit Script II. Bei Script 1.0 von 1990 (abzulesen an der Info-Dialogbox) ist nur die nachträgliche Trennung mit ScripTipp oder mit der Maus möglich.

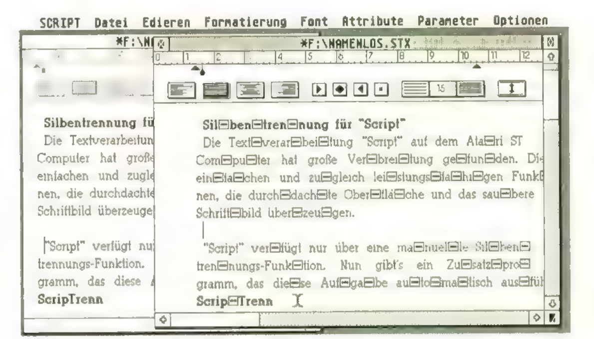
Der saubere Randausgleich durch die Silbentrennung macht Texte schöner und lesbarer. Da mehr Text auf die Zeilen paßt, wird etwa 1% Papier weniger verbraucht. Zeit läßt sich bekanntlich nicht einsparen - doch die Accessories 1st Trenn und ScripTrenn können eine nervtötende Routinearbeit automatisieren. Wer eine Zeitlang mit solch einer sicheren Silbentrennung gearbeitet hat, wird auf sie nicht mehr verzichten wollen.

Oliver Völckers

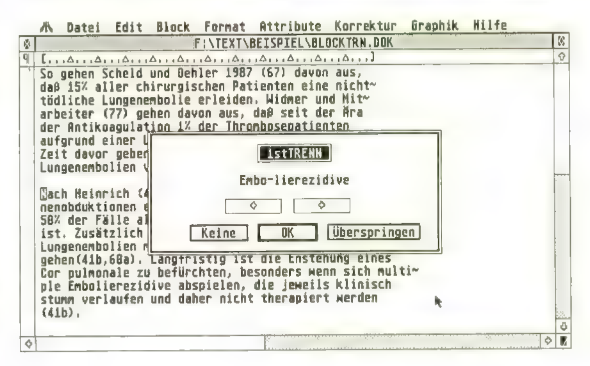
Bezugsadressen:

APiSoft
Bundesallee 56
1000 Berlin 31
Tel. 030 / 853 43 50

MAXON Computer
Schwalbacher Straße 52
6236 Eschborn
Tel. 0 61 96 / 48 18 11



Die Trennhilfszeichen von Script



1st Trenn in Wordplus

ROCKUS





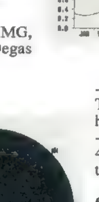


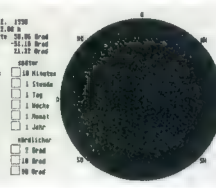
Skyplot Plus 3c

Astronomieprogramm der Spitzenklasse

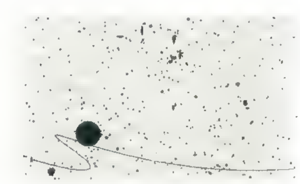
LEISTUNGSDATEN:

- bis zu 64.000 Sterne
- bis zu 32.000 Nebel etc.
- umfangreiche ON-Line Hilfen
- Echtzeitsimulation
- jede Menge Diagramme und Simulationen zur Verdeutlichung von astronomischen Sachverhalten
- Bilder laden, speichern : IMG, AIM, STAD, Neochrome, Degas (Elite) komprimiert.





- eingebaute DOS-Funktionen: Dateien löschen, Formatieren und Programme starten
- umfangreiche Parameter für Grafik und Drucker
- Zeichnen, Speichern und Laden von Bewegungsbahnen
- Plottausgabe auch für HPGL-kompatible Plotter
- Laden von Daten stark beschleunigt



- Finsterniskanon
- Benutzung der Hardwareuhr
- Zonenzeit und Zeitzonen
- 3D-Darstellung auch für Stereoskope
- Grafiken bearbeiten: Spiegeln, Beschriften etc.
- Speichern von Sequenzen

- eingebaute Editoren für Textdateien und Sternbildhilfinien
- Bitmaptreiber mit 6912 X 4320 Pixel sowie Hardcopytreiber für 24 und 48-Nadler

Skyplot gibt es als

N O R M A L-
oder als
C O - Prozessor

Ausführung und zwar ab sofort

ohne Kopierschutzmodul

Skyplot Normalv.	198,--DM
Skyplot Co-Proz.	248,--DM
Up-Date Skyplot auf Co-Prozessor	50,--DM
Up-Date von alten Skyplot-Versionen auf Version 3C	50,--DM
auf Co-Proszessor	100,--DM

Zum Schluß noch einige Pressestimmen zu Skyplot Plus 3c::

... goldenes Byte für Skyplot (aus Computer Persönlich)

... ein unglaubliches Programm ... einfach und eindrucksvoll...
(ATARI Magazin)

ST-Inventur

einfache Erfassung der jährl. Bestandsaufnahme

Das lästige Unterfangen der alljährlichen Bestandsaufnahme in den Unternehmen und Kleinbetrieben, soll mit Hilfe des Programmes "ST-INVENTUR" erleichtert werden. Das Programm ist aus dem Eigenbedarf heraus entstanden und wird schon seit mehreren Jahren eingesetzt.

Programmaufbau:
Die Verwaltung der Bestandsgüter erfolgt hierachisch in vier Ebenen:
- Geschäftsbereiche
- Firmen
- Sortimente
- Artikel

Auf der untersten Ebene werden maximal 99 Artikel zu einem Sortiment zusammengefaßt. Dabei enthält ein Artikeldatensatz die Informationen:
- Artikelname
- Artikeleinheit: Stück, Meter, Kilogr., Liter
- Artikelmenge
- Artikelpreis

iederum 99 Sortimente sind in einer Firmendatei vereint.Max. 99 Firmen können in einem Geschäftsbereich untergebracht werden. Die tatsächliche Anzahl der speicherbaren Bestandsmenge ist von der Größe des verwendeten Speichermediums, wie z.B. Diskette oder Platte abhängig.

Funktionsumfang:
- Inventurliste bearbeiten
- Inventuraufnahmeliste drucken
- Inventurliste drucken
- Änderungen der Inventurgrunddaten
- Löschen der Inventurmengen und/oder der Inventurpreise
- Neuberechnung der Inventursummen
- Statistische Auswertung

Bedienung:
- vollständig menügeführt
- vier Arbeitsebenen stehen zur Verfügung:
  neue Daten eingeben/bestehende Daten ändern/Anwahl eines Datums etc.

```
SPEZIELLE FUNKTIONEN
====================

-0-  Hauptmenü
-1-  Änderung der Grunddaten
-2-  Löschen der Mengen und/oder Preisen
-3-  Neuberechnung der Inventursummen
-4-  Statistische Auswertung
```

```
Statistische Auswertung
=======================

Die Jahresinventur von Firma Computer Chaos
von Geschäftsjahr 1981 beinhaltet:

 5 Geschäftsbereiche ; davon  4 unbearbeitet
20 Firmen ; davon  18 unbearbeitet
41 Sortimente ; davon  38 unbearbeitet
28 Artikel ; davon  8 unbearbeitet
```

ST-Inventur gibts für DM 498,--

Preise sind unverbindlich empfohlene Verkaufspreise

BESTELL - COUPON

Heim Verlag

Heidelberger Landstr.194
6100 Darmstadt-Eberstadt
Telefon 06151 - 56057
Telefax 06151 - 56059

Bitte senden Sie mir

____ Skyplot ❑ Normalv. ❑ Co-Proz. a 198,--/248,-- DM zzgl. Versandk. DM 6,--
____ ST-Inventur a 498,-- DM (Ausland DM 10,--)
unabhängig von der bestellten Stückzahl

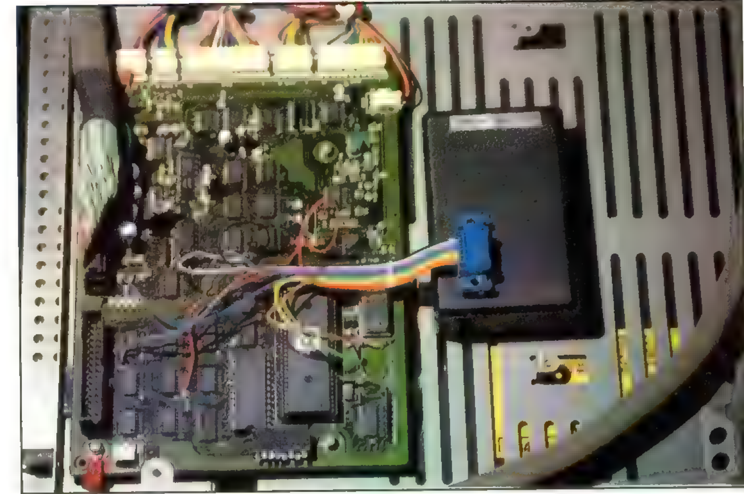
Name, Vorname ____________
Straße ____________
PLZ, Ort ____________
oder benutzen Sie eingeheftete Bestellkarte

in Österreich:
RRR EDV GmbH
Dr. Stumpf Str. 118
A-6020 Insbruck

in der Schweiz:
Data Trade AG
Landstr.1
CH-5415 Rieden-Baden

Erweiterter Horizont

600 dpi mit dem ATARI-Laserdrucker



Alles das, was von einem Computerdrucker irgendwie einmal zu Papier gebracht werden soll, setzt sich aus einzelnen Punkten zusammen. Bei einem 9- bzw. 24-Nadeldrucker ist das auch sehr deutlich zu erkennen. Gerade im Schnellschreibmodus (Draft) stört viele Benutzer die Deutlichkeit der Nadelabdrücke. Durch verschiedene Tricks kann der Nadeldrucker aber überlistet werden, so daß die Buchstabenkonturen geradliniger herauskommen.

Selbst wenn es zunächst merkwürdig klingen mag, auch ein Laserdrucker muß Grafik und Schrift aus einzelnen Punkten zusammenbauen. Nur durch die Art der Aufbereitung scheinen die häßlichen Pünktchen verschwunden zu sein, es werden die Übergänge von Punktdefinition zu Punktdefinition fließender. Die Größe und Anzahl der Punkte bestimmen, wie fein die Ausfranzungen an den Buchstabenkanten werden. Gleichzeitig kommt dem Schriftbild eine Eigenschaft des menschlichen Auges zugute: Bei weiter entfernt gehaltenem Papier wird der Ausdruck immer undeutlicher, die Treppenlinien, z.B. bei einem Kreis, sind nicht mehr sehr klar auszumachen. Dennoch wird das geübte Auge solche Feinheiten finden, die sich beispielsweise bei einem repräsentativen Brief oder bei einer Druckvorlage störend bemerkbar machen.

SLM 804 und mehr

Der ATARI-Laserdrucker SLM 804 hat eine Auflösung von 300 dpi („dpi" heißt „dots per inch", zu deutsch: „Punkte pro Zoll - 1 Zoll beträgt 25,4 mm), und zwar sowohl horizontal als auch vertikal. Somit besteht eine komplette DIN A4-Seite aus 2400 * 3386 Punkten, die natürlich nicht alle definiert sind, sonst wäre das Blatt schwarz.

Um nun die Konturen von Rundungen (wer kennt die Treppeneffekte bei gebogenen Linien nicht?), aber auch von relativ kleiner Schrift noch feiner werden zu lassen, müßte man versuchen, einfach mehr Bildpunkte für die gleiche Flächengröße zu verwenden. Das analoge Abbildungsprinzip des Laserdruckers kommt uns bei einer bestechend einfachen Idee dabei entgegen: Ein Laserstrahl wird mit drei rotierenden Spiegeln in horizontaler Richtung abgelenkt und überstreicht dabei mit Unterbrechungen Zeile für Zeile von links nach rechts auf einer rotierenden Trommel. Diese Trommel hat eine extrem glatte Oberfläche und wird genau an jenen Stellen, wo der Laserstrahl sie berührt, also wo später ein Bildpunkt erscheinen soll, elektrostatisch aufgeladen. Diese Ladungsträger könnten im äußersten Fall die Größe nur eines Elektrons haben, treten aber aufgrund der relativ großen Streuung des Laserstrahls als regelrechte Ladungsbündel auf. Später streift ein Partikelbesen mit negativ geladenen (schwarzen) Farbteilchen (genannt Toner) über die Trommel, so daß diese Partikel genau dort auf der Trommeloberfläche hängenbleiben, wo die positiven Ladungsträger sitzen - Gegenpole ziehen sich an (siehe Physikstunde bei Lehrer Lämpel). Als letzter Arbeitsschritt wird ein vorgeheiztes Blatt Papier um die Trommel gelegt, weiter aufgeheizt, gleichzeitig ändert sich die Polung der Trommel in negativ und stößt damit alle Tonerpartikelchen ab. Schließlich ist der Toner in das Papier eingebrannt.

Es ist nun überhaupt kein Problem, dem Laserstrahl einfach genau die doppelte Anzahl an Punktdefinitionen (pro Zeile) in horizontaler Richtung mitzugeben. Spiegelmechanik und Plazierungsgenauigkeit des Laserstrahls sind durchaus in der Lage, hierfür noch präziser zu arbeiten. Das Problem liegt nur in der doppelten Menge an Punktdefinitionen. So muß der Steuerelektronik mitgeteilt werden, daß nicht 2400 sondern 4800 Bildpunktinformationen einer Druckzeile entsprechen. Auch muß der Laserstrahl entweder doppelt so langsam über die Trommelzeile geführt werden (was ein weiteres mechanisches Problem aufwerfen würde), oder er müßte die doppelte Informationsmenge in derselben Zeitspanne verarbeiten (was ohne weiteres verwirklicht werden kann). Wie leicht nachzuvollziehen ist, wird die

logische Größe der Bitmap in horizontaler Richtung verdoppelt und braucht demzufolge auch die doppelte Menge an Speicherplatz im RAM. Gleichzeitig muß das spätere Druckbild wieder auf die physikalische Größe eines DIN A4-Blattes gestaucht werden.

Aufgrund des beschriebenen Arbeitsprinzips ist nun auch einzusehen, daß mit denkbar einfachen Mitteln eine Verdoppelung der Auflösung in horizontaler Richtung leicht zu verwirklichen ist. Alle beschriebenen Aufgaben erfüllt ein kleines Zusatzgerät, das „600-dpi-Kit" der Firma Marvin AG aus der Schweiz.

Um der Wahrheit die Ehre zu geben: Die 600 dpi erreicht dieses Zusatzgerät nur in horizontaler Richtung, also 600 * 300 dpi. In vertikaler Richtung sieht das Ganze wieder anders aus. Die Trommel dreht sich in konstanter Geschwindigkeit so langsam, daß die Zeit für den Laserstrahl ausreicht, die Zeileninformationen vollständig auf der Trommel aufzutragen. An die Verdoppelung der Zeilenzahl hat man sich noch nicht herangewagt, weil dies einen höheren mechanischen Aufwand erfordern würde.

Der Umbau

Um es gleich vorwegzunehmen, man sollte das Arbeiten mit einem Lötkolben auf jeden Fall beherrschen und ein kleinwenig elektrotechnische Grundkenntnisse besitzen, bevor es zur Sache geht. Sonst wäre der Fachhändler wohl die bessere Adresse für den Umbau. Wenn Netz- und DMA-Kabel abgetrennt sind, wird einfach der obere Gehäusedeckel abgehoben. Unter einem Schutzblech verbirgt sich die Platine mit der Steuerelektronik. Dort ist auch noch ausreichend Platz für unser 600-dpi-Kit.

An zwei Stellen müssen die Leiterbahnen durchtrennt werden. Das ist etwas Kleinarbeit und mit Vorsicht zu bewerkstelligen, damit die Nachbarbahnen nicht getroffen werden. Am besten nimmt man einen spitzen Gegenstand (Schraubenzieher oder Dörnchen) und kratzt die Leiterbahn an einer leicht zugänglichen Stelle durch. Das mitgelieferte Handbuch zum 600-dpi-Kit besitzt einen Mittelteil mit vier Farbfotos, die die günstigen Stellen sehr deutlich zeigen. Das Zusatzkästchen hat nun neun farbig gekennzeichnete Verbindungsleitungen, die unter anderem an die Lötbeinchen einzelner ICs angebracht werden müssen. Das ist nicht sehr einfach, weil man bei den eng aneinanderliegenden Beinchen sehr schnell das Nachbarbein trifft. Auch die anderen Lötpunkte, besonders die durchkontaktierten Bohrungen, sind sehr klein ausgefallen und außerdem mit Lötstoplack überzogen. Da muß man zuerst vorsichtig den Lack abkratzen und dann genauso vorsichtig löten. Alles in allem ist es schon eine Tüftelarbeit, die wirklich nur ein Elektronikbastler oder ein Fachmann ausführen sollte. Wegen der deutlichen Fotos im Handbüchlein sowie der sehr ausführlichen Lagebeschreibung wird es sicher keine Probleme geben, die richtigen Lötpunkte zu finden.

Ende gut ...

Natürlich muß nach der mühevollen Arbeit eine Erfolgskontrolle stattfinden (hoffentlich mit einem Erfolgserlebnis). Man sollte den Drucker noch nicht vollständig verschrauben, sondern einfach den Hauptdeckel aufsetzen und das Testprogramm durchführen.

Und das ist auch schon ein kleiner Wermutstropfen. Die Entwickler dieses 600-dpi-Kits sind davon ausgegangen, daß sich wahrscheinlich nur CALAMUS-Besitzer für dieses Zubehör interessieren. Deswegen sind verschiedene Testdokumente und zwei Druckertreiber lediglich für das Programm CALAMUS auf Diskette vorhanden. Was machen alle diejenigen, welche CALAMUS nicht besitzen? Hier wäre vielleicht ein eigenes kleines Testprogramm angebracht. Was kann sonst noch das Schriftbild verschlechtern?

1. könnte eine schon in die Jahre gekommene Trommel unseres Laserdruckers schuld sein, wenn das Druckergebnis nicht befriedigt. Die Tonerkörnchen werden bei 600 dpi feiner verteilt, und die Trommel muß in der Lage sein, diese auch entsprechend präzise plaziert festzuhalten. Alte Trommeln weisen in der Regel typische Merkmale auf: Sie sind entweder so stark vormagnetisiert, daß sie auch an unerwünschten Stellen Toner aufnehmen: das Resultat sind Tonerspuren auf weißen Flächen, oft an Rändern. Oder es kann passieren, daß die Trommel nicht mehr ausreichend statisch auflädt, dann erscheint das Druckbild blaß und dünn, und schwarze Flächen sind nicht mehr tiefschwarz.
2. kann das Papier zu einem guten Schriftbild beitragen. Zu rauhe Oberflächen nehmen die jetzt feinere Körnung nicht mehr exakt an. Im Handel gibt es sogar spezielles 600-dpi-Papier, dessen Oberfläche gleichmäßiger und glatter ist.
3. gibt es speziellen, feinkörnigen 600-dpi-Toner zu kaufen, der allerdings um einiges teurer ist. Der Unterschied zu dem handelsüblichen Toner ist nicht sehr gravierend - und wer hat ständig hochwertige 600-dpi-Ausdrucke?
4. müßte zu prüfen sein, ob die Leuchtstärke der Laserdiode eine Verbesserung erreichen kann. Ist die Laserdiode zu hell eingestellt, werden dünne gerade Linien überstrahlt und erscheinen zu dünn bis überhaupt nicht. Ist der Laser zu dunkel eingestellt, erscheinen schwarze Flächen leicht grau und decken nicht mehr. Zum Abgleich dieses Problems sollte man in einem beliebigen Grafikprogramm eine dünne schräge Linie (vielleicht auch mehrere mit unterschiedlicher Strichstärke) zeichnen und ein vollschwarzes Rechteck dazu.

Die liebe Software

Der Befehl „Einzelblatt" von einem Druckertreiber aus muß ab sofort leider entfallen, weil das „Manual-Feed-Bit" jetzt zur Umschaltung zwischen 300 und 600 dpi benutzt wird. Also: Wenn Sie künftig den Ausdruck per Einzelblattschacht angeben, ist das gleichbedeutend mit 600 dpi, und wenn Sie den Ausdruck aus dem Magazin ansteuern entspricht dies sofort 300 dpi.

Die vermeintliche Einschränkung, daß nun nicht mehr von der Software bestimmt werden kann, ob per Einzelblatt oder per Papierkassette ausgedruckt wird, ist in Wirklichkeit kein Nachteil. Zwar ist die Verbindung vom Einzelblattdetektor aufgetrennt, und das Signal, daß ein Blatt Papier in der Einzelzuführung liegt, wird nicht mehr zum Anwenderprogramm weitergeleitet, aber das 600-dpi-Kit wertet diesen Umstand intern aus und veranlaßt bei vorhandenem Papier in Einzelzuführung, daß dieses bevorzugt verwendet und der Magazinschacht nicht benutzt wird. Da kann es durchaus Verständigungsprobleme mit alten Druckertreibern geben.

Wie schon beschrieben, ist unsere Bitmap bei 600 * 300 um das Doppelte größer, was natürlich auch einen größeren RAM-Speicher beansprucht. Auch dies muß die Software entsprechend bedienen.

Was ist mit GDOS, AMCGDOS, FSM-GDOS? Als Benutzer merken Sie die höhere Auflösung erst auf dem Papier, weil alle GDOS-Treiber unterschiedliche Fonts für Bildschirm und Ausdruck benutzen. Die Bitmap im Speicher hat unter GDOS die üblichen 2336 * 3386 Bildpunkte (entspricht 300 dpi). Erst beim Ausdrucken wird vom GDOS-Treiber auf 600 dpi umgerechnet. Da diese Methode leider nicht immer die besten Ergebnisse bringt, wäre das Verwenden spezieller Fonts (auch angepaßter) wohl sinnvoller. Wer bietet 600*300-GDOS-Fonts gegen eine Briefmarkensammlung?

Der FX-Emulator (z.B. „Laserbrain“) nutzt die höhere Auflösung unter Verwendung spezieller Fonts. Die alten Fonts sind unter 600 dpi nur eingeschränkt zu empfehlen, weil die Buchstaben horizontal gestaucht sind (sieht lustig aus). Tausche Wellensittich gegen Laserbrain-Fonts!

Wer eigene Druckertreiber schreiben kann (oder jemanden kennt, der weiß, wo jemand wohnt, der eigene Druckertreiber schreiben kann) ist fein raus: Sprechen Sie den Laserdrucker in Ihrer Software direkt an (Sie Glücklicher), so müssen Sie lediglich für 600 dpi beim „MODE_SELECT-Befehl“ das „manual-feed-Bit“ setzen und beim Drucken eine 4800 * 3386 große Bitmap übergeben. Das größte Problem dürfte sein, in Ihrer Software der Punktgröße Rechnung zu tragen, da die meisten Programme von der Annahme ausgehen, daß die Punkte quadratisch oder rund sind. Bei 600 * 300 Pixel sind die einzelnen Punkte immer doppelt so hoch wie breit. Bei punktorientiert arbeitender Software müssen Sie das Bild auf interpolativem Wege anpassen. (Interpolation ist ein Annäherungsverfahren in der Mathematik.) Vektororientierte Software hat bekanntermaßen mit der Punktgröße überhaupt keine Schwierigkeiten. Wer kennt jemanden, der jemanden kennt ...

Nun das Schlußwort

Die Werbung ist schon lustig: „Noch kein Farbdrucker, aber ganz nah dran ...“ (kein Kommentar). Wie dem auch sei, für alle diejenigen, die professionell mit dem ATARI-Laserdrucker SLM 804 arbeiten wollen und für die eine höhere horizontale Auflösung wichtig ist, stellt dieses Zubehör der Firma Marvin AG eine sinnvolle Erweiterung dar. Der Preis von 410 DM ist nicht überteuert, wenn man bedenkt, um wieviele Hunderterstellen ein wirklicher 600-dpi-Laser mehr kosten würde als unser SLM 804 mit ca. 2500 DM.

DK

Bezugsquelle:

H. Richter
Distributor
Hagener Straße 65
5820 Gevelsberg
Telefon: 02332 / 2706

OMIKRON Bücher,

die Literatur zum ATARI-Basic



Merkmale Band I:
- zeigt anhand vieler Beispiele die Entwicklung von Programmen
- Vorstellung wichtiger Unter- und Hilfsprogramme
- Einsatz von GEM-Funktionen in OMIKRON-Basic
- die grafischen Fähigkeiten des Computers werden anhand von Beispielen erklärt
- Anwendungen aus dem Bereich der Mathematik

INHALT Band I:
- Druckeranpassung
- Feststellen der Existenz einer Datei
- verwenden mehrerer Bildschirme
- GEM-Programmierung
- Hilfsprogramme
- Editor für Mauszeiger
- Turtle Grafik
- Integration nach Simpson
- Spiele
- Dreidimensionale Objektdarstellung
- Kopieren mehrerer Files
- Arbeiten mit Fenstern, Dialogboxen
- Adressverwaltung
- Ermitteln von Mittelwerten, Varianz und Standardabweichung

B-411
HARDCOVER
incl. Programmdiskette
ISBN 3-923250-55-X

DM 59,--

MERKMALE Band II:
- neue Tips und Tricks
- Benutzerführung mit mehreren Menütypen
- Präsentation neuer Spiele
- Simulation einer Touring-Maschine
- umfangreiche Turtlegrafik-Library
- Einbindung von Assembler Routinen in Basic
- der mit den Grundlagen der Basic-Programmierung
- der Mit den Grundlagen der Basic-Programmierung vertraute Leser findet neben zahlreichen fertigen Programmen, die ausführlich gestestet wurden, viele Tips und Anregungen für die Anwendung in eigenen Projekten.



INHALT Band II:
- Hilfsroutinen
- Menüverwaltung
- ISAM-Dateiverwaltung
- ASSEMBLER-Einbindung
- Disk-Utilities
- 3D-Programmierung

B-440
HARCOVER
incl. Programmdiskette
ISBN 3-923250-82-7

DM 59,--

INHALT:
- Erklärung der Schleifen- und Programmstrukturen - Primzahlenberechnung - Zahlenraten
- Numerische und Stringfunktionen, alle trigonometrischen Funktionen wie Sinus, Cosinus
- Unterprogramme und Prozeduren - Rekursive Prozeduraufrufe
- Variablentypen und Arrays - Sieb des Eratosthenes - Adresseingabe
- Grafikprogrammierung - Blockoperationen Grafische Grundelemente (Kreis, Linie, Quadrat)
- Multitasking in Omikron Basic - Druckerspooler
- Programmieren von abstrakten Datentypen Verkettete LIsten - die Datenstrukturschlange
- Dateiverwaltung (sequentiell, Random Access)
-Programmprojekt Fakturierrung ist auf ca. 30 Seiten ausführlich beschrieben
-Programmierung des Betriebssystems (Einsatz von Alertboxen, Pull-Down Menüs)
- Programmieren einer Druckeranpassung in OMIKRON-Basic
- Programmentwicklung und Debugging
- Fehlersuche und Beseitigung
- Sammlung von vielen Beispielprogrammen, die sich auf der mitgelieferten Programmdiskette befinden - Ausgabe eines Diskettenverzeichnisses - Back-Up Programm für die Festplatte - Turtle Grafik in Omikron Basic
- Adressverwaltung
- das Buch enthält eine alphabetisch geordnete Übersicht über alle Befehle

B-413
Hardcover
über 500 Seiten
ISBN 3-923250-60-6
incl. Programmdiskette

59.-- DM



BESTELL - COUPON

Bitte senden Sie mir

____ Programmieren in Omikron Basic Band 1 a 59,-- DM
____ Programmieren in Omikron Basic Band 2 a 59,-- DM
____ Das Große Omikron Basic Buch a 59,-- DM

zzgl. Versandk. DM 6,-- (Ausland DM 10,--) unabhängig von der bestellten Stückzahl

Name, Vorname ____________________

Straße ____________________

PLZ, Ort ____________________

oder benutzen Sie eingeheftete Bestellkarte

Heim Verlag

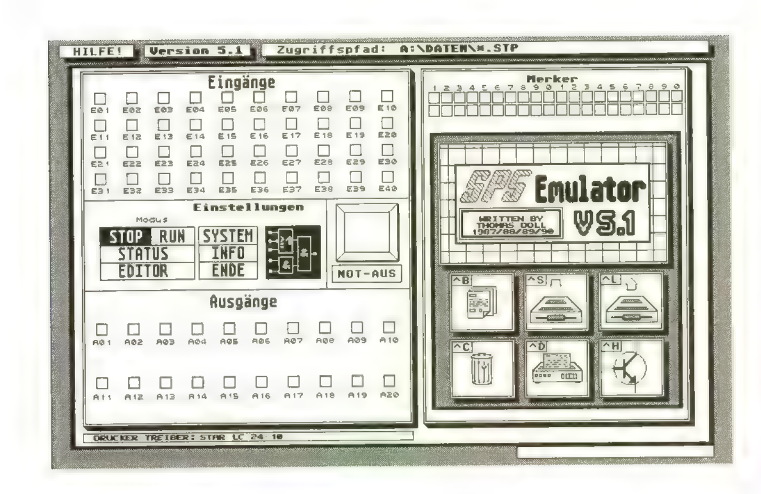
Heidelberger Landstr.194
6100 Darmstadt-Eberstadt
Telefon 06151 - 56057
Telefax 06151 - 56059

in Österreich:
RRR EDV GmbH
Dr. Stumpf Str. 118
A-6020 Insbruck

in der Schweiz:
Data Trade AG
Landstr.1
CH-5415 Rieden-Baden

SPS

Speicher-programmierbare Steuerung



Seit ungefähr 20 Jahren, und in letzter Zeit immer stärker, geistert ein Begriff durch unsere Industrie. „Speicherprogrammierbare Steuerung“ heißt dieses Zauberwort (im folgenden kurz „SPS“ genannt), das viele Menschen in Entzücken versetzt, manchen Leuten aber auch etwas Unsicherheit einflößen kann. Dieser Artikel soll dazu dienen, Ihnen zu erklären, was SPS eigentlich ist, was man damit machen kann, worin die Vorteile liegen und schließlich: Wie kann ich mit meinem Atari-Computer die SPS-Programmierung kennenlernen? Der folgende Artikel und alle technischen Daten beziehen sich auf eine Siemens-STEP-5-SPS-Anlage, da diese doch sehr weit verbreitet und mittlerweile, besonders in der Ausbildung, zur Norm geworden sind.

Allgemeines

In unserer heutigen automatisierten Industrie setzt sich immer mehr die Elektronik zur Kontrolle von komplizierten Fertigungsvorgängen durch. Automatisierungsaufgaben werden dadurch mit immer kleineren, leistungsfähigeren und billigeren Anlagen gelöst. Wenn man bedenkt, mit was für einem Aufwand an Sicherungseinrichtungen (Schütz), Hilfsrelais und sonstigen Steuerungsbauteilen früher eine Steuerung aufgebaut wurde und daß sich diese Steuerung mit einer programmierbaren Steuerung komplett ersetzen läßt, so wird einem sehr schnell klar,daß diese Art der Steuerungstechnik Zukunft hat. Die SPS, was ja „speicherprogrammierbare Steuerung“ heißt, zeichnet sich durch sehr große Flexibilität und Erweiterbarkeit aus. Will man zum Beispiel den Prozeßablauf einer SPS-gesteuerten Bandstraße ändern, so steckt man das passende EPROM (löschbares, programmierbares ROM) in den Steckplatz - und die Bandstraße ist fertigungsbereit.

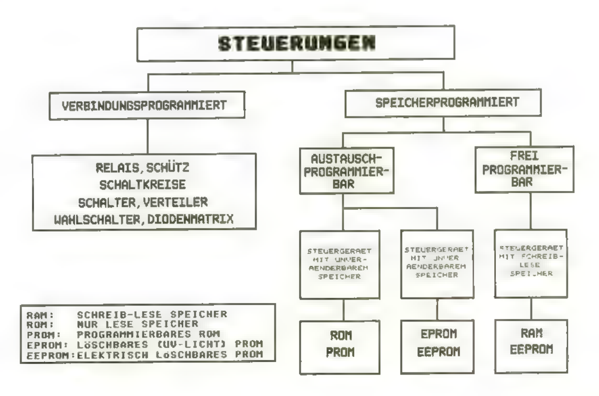
Doch nun ein wenig zur Entwicklungsgeschichte der SPS: Speicherprogrammierbare Steuerungen werden seit 1970 in der Technik verwendet. Die ersten Anlagen wurden in Amerika eingesetzt und schon bald danach kam der neue Trend auch zu uns nach Deutschland. Die ersten Anlagen hatten noch mit Kinderkrankheiten zu kämpfen, die aber in den folgenden sechs Jahren ausgemerzt wurden. Zu den größten Problemen zählten Erschütterung, Temperaturschwankungen, Störimpulse, Spannungsspitzen und vieles mehr; aber mit der Zeit bekam man dies alles in den Griff und entwickelte eine robuste und fehlerfrei arbeitende Hardware. Heute verwendete Steuerungen basieren auf dem neuesten Stand der Mikroprozessortechnik und sind damit dementsprechend leistungsfähig.

Was ist eine Steuerung?

Die Aufgabe einer elektrischen Steuerung liegt darin, Einzeloperationen im Arbeitsprozeß einer Maschine oder Anlage nach einem festgelegten Funktionsablauf in Abhängigkeit vom Maschinenzustand und dem Zustand der Befehlsgeber, zu leiten. Die Steuerungen kann man nach verschiedenen Kriterien unterscheiden:

Art der Signalverarbeitung

- Verknüpfungssteuerung
  Eingangssignale werden als logische Zustände (HIGH/LOW) erfaßt, und nach bestimmten Verknüpfungsoperationen (UND, ODER, NICHT) er-



folgt die Reaktion am Ausgang der Steuerung.
- Ablaufsteuerung
  Der Steuerungsprozeß läuft schrittweise ab, wobei das Weiterschalten zum nächsten Programmschritt abhängig von den Weiterschaltbedingungen ist. Diese Weiterschaltbedingungen können zeit- oder prozeßabhängig sein.
- Asynchrone Steuerung
  Direkte Verarbeitung der Eingangssignale ohne zusätzliches Taktsignal (langsame Steuerungen im Starkstrombereich)
- Taktsynchrone Steuerung
  Synchron mit einem Taktsignal laufende Verarbeitung der Signale (sehr schnelle Steuerungen)

Art der Programmverwirklichung
- VPS (verbindungsprogrammierte Steuerung)
  Die Steuerung wird durch feste Drahtverbindungen unter den einzelnen Steuerungsgeräten (Sicherungseinrichtungen, Relais) FEST programmiert, d. h. eine eventuelle Änderung des Programmes zieht eine teilweise oder komplette Neuverdrahtung nach sich. Zu diesen Steuerungen kann man alle verdrahteten Schutzsteuerungen zählen. Bei Erstellung einer VPS müssen die Steuerungsbedingungen bekannt sein.
- SPS (speicherprogrammierte Steuerung)
  Bei der SPS liegt die Steuerung als Programmcode im Arbeitsspeicher der Steuerungs-Hardware vor. Falls dieser Arbeitsspeicher ein Schreib-/Lesespeicher (RAM) ist, kann der Programmcode mittels einer Tastatur (Programmiergerät) geändert werden (frei programmierbar). Ist der Arbeitsspeicher ein Nur-Lesespeicher (ROM), so kann ein anderes ROM, das den neuen Programmcode trägt, eingesetzt werden, (austauschprogrammierbar). Jede SPS kann an ihr Anwendungsgebiet angepaßt werden. Deshalb existieren Grundmodule, die man dann für die jeweiligen Anforderungen anpaßt, das führt zu sehr großer Flexibilität und geringere Typenvielfalt, was niedrigere Preise durch mögliche Massenproduktion bringt.

Art der Aufbauorganisation
- In einem Betrieb können sehr viele Einzelsteuerungen nötig sein, damit der Fertigungsprozeß vom Rohmaterial bis zum verpackten Endprodukt abläuft. Eine EINZELSTEUERUNG ist die Steuerung einzelner Stellglieder. Es gibt nun die Möglichkeit, die Einzelsteuerungen eines Teilprozesses in einer sogenannten GRUPPENSTEUERUNG zusammenzufassen. Diese Gruppensteuerung sitzt über den Einzelsteuerungen und koordiniert deren Einzelprozesse. Die Gruppensteuerungen werden mit LEITSTEUERUNGEN zum Gesamtprozeß zusammengefaßt. Es können also regelrechte Steuernetze mit parallel ablaufenden Teilprozessen aufgebaut werden. Somit ist es möglich, über ein gut aufgebautes Netzwerk den gesamten Fertigungsprozeß flexibel und vollautomatisch zu steuern. Hauptsächlich werden Parallel- oder Master/Slave-Systeme verwendet.

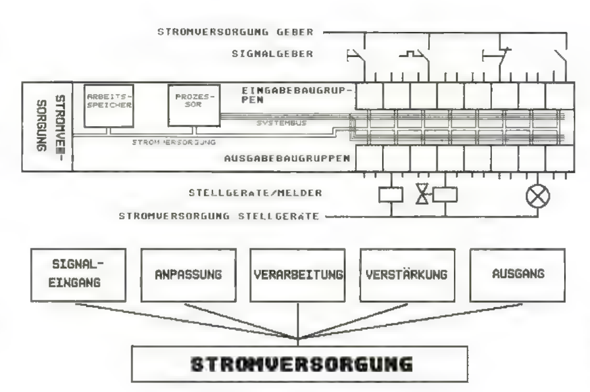


Bild 2: Übersicht über den Grundaufbau eines SPS-Gerätes

Vorteile der SPS gegenüber anderen Steuerungsmethoden

Die SPS zeichnet sich durch eine große Zahl an Vorteilen gegenüber der herkömmlichen Steuerungsmethode, der VPS, aus:

- Die SPS-Geräte sind standardisiert.
- Programmänderungen während der Installation und auch während des Betriebes sind einfach und schnell durchzuführen.
- Der Raumbedarf für eine VPS-Steuerung ist je nach Steuerungsproblem sehr groß. Die gleiche Steuerung mit einer SPS-Anlage benötigt relativ zur VPS sehr wenig Platz, da keine Hilfsschutzeinrichtungen und sonstige Steuerorgane nötig sind.
- Die Duplizierung, Archivierung und Auflistung eines SPS-Programmes ist eine Sache von Minuten.
- schnelle und sichere Archivierung auf Datenträgern
- Weniger Montagearbeit, weniger Materialbedarf und die Möglichkeit der Parallelentwicklung eines Programmes von mehreren SPS-Programmierergruppen machen eine SPS-Projektierung mit möglichst wenig Zeitaufwand möglich.

Es ist ganz offensichtlich, daß man mit der SPS eine wesentlich flexiblere und zeitsparendere Steuerungsmethode entwickelt hat. Des weiteren gilt es zu bedenken, daß der heutige Stand der Elektronik einige sehr wichtige Möglichkeiten für die Optimierung von Wartung, Produktion und Effektivität bieten kann: semiintelligente Systeme, die dem Monteur Fehleranalysen und Wartungsvorschläge machen (ich bin mir sicher, daß es irgendwann auch einmal Systeme gibt, die in beschränktem Umfang fähig sind, sich selbst zu warten), ergonomische Vorraussagen treffen und somit zur Nutzung mit optimalem Wirkungsgrad beitragen und selbständig auf Notsituationen zum Schutz der Anlage und ihrer Bediener reagieren.

Aufbau einer SPS

Eine speicherprogrammierbare Steuerung besteht aus einem Automatisierungsgerät das die Mikroprozessortechnik zur Abarbeitung des SPS-Programmes enthält. Das Automatisierungsgerät (kurz: AG) verfügt über Eingänge, an denen die peripheren Signalgeber (Endschalter, Taster, Näherungsschalter) angeschlossen werden. Das AG verfügt auch über Ausgänge, an denen Stellgeräte (Lastschutzeinrichtungen, Magnetventile, Leuchten) angeschlossen werden. Ein AG besteht im wesentlichen aus einem Stromversorgungsnetzteil, Ein- und Ausgabegruppen und der Zentraleinheit mit Steuerwerk und Programmspeicher. Für die Realisierung von Zeitgliedern sind auch noch bestimmte Zeitbaugruppen integriert.

Die Signalgeber liefern ihr Zustandssignal an die Einbaugruppen. Diese Signale werden über den internen Systembus an die Zentralbaugruppe geliefert. In der Zentraleinheit bearbeitet das Steuerwerk das im Speicher stehende Programm und verknüpft dabei die logischen Zustände (HIGH/LOW) der Eingangsbaugruppen, indem es auf den Systembus zugreift. Abhängig von diesen logischen Zuständen und den im SPS-Programm erfolgten logischen Verknüpfungen erfolgen dann die Reaktionen an der Ausgangsbaugruppe, an die die Stellgeräte (Schutzeinrichtungen, Ventile) angeschlossen sind. Die Ausgangsbaugruppe bekommt ihre Zustände über den Systembus vom Steuerwerk zugewiesen. Die Stromversorgungsbaugruppe liefert die nötigen Betriebsspannungen für das AG (5V TTL) und 24 V Schutzkleinspannung für die Geber und Stellgeräte. Geber und Stellgeräte werden meist mit 24 V oder 220 V betrieben und von einem separaten Netzteil oder Steuertransformator gespeist. Bild 2 soll eine klare Übersicht über den Grundaufbau eines SPS-Gerätes liefern.

Signaleingabe: Sie erfolgt durch jede Art von Signalgeber wie Taster, Endschalter, induktive/kapazitive Näherungsschalter. Die Signale können je nach Steuerung digital oder analog erfaßt werden.

Anpassung: Eine Anpassung wird benötigt, wenn der Spannungspegel oder die Signalart des Gebers nicht kompatibel zur Eingangsbaugruppe sind. Beispiele für eine Anpassung stellen Spannungswandler dar. Eine galvanische Trennung (Optokoppler) kann auch in der Anpassung erfolgen.

Verarbeitung: In diesem Teil der SPS-Anlage werden alle logischen Verknüpfungen und Verriegelungen , Zeitabläufe, Speicher- und Zählfunktionen realisiert. Der Verarbeitungsteil ist das hauptsächliche Arbeitsgebiet der Steuerungsysteme. In der VPS kommen hier Hilfskontakte, Zeitrelais und Hilfsrelais, in der elektronischen Steuerungstechnik Schaltkreise (wie UND, ODER, NICHT usw.), SPS oder Prozeßrechner zum Einsatz.

Verstärkung: Hier werden die Signale aus dem Verarbeitungsteil auf ein solches Leistungsniveau verstärkt, daß die Stellgeräte angesteuert werden können.

Ausgabe: Hier sind die Stellglieder angeordnet, die direkt den Prozeß beeinflussen (Thyristoren, Schutzeinrichtungen, Magnetventile).

Wie programmiere ich eine Steuerung auf einer SPS?

Zuerst ein kleiner Exkurs zu unserem ATARI ST. Wußten Sie, daß es einen SPS-Emulator in der Sonderdisksammlung der Firma MAXON Computer GmbH, Eschborn gibt? Für 25 DM ist es Ihnen möglich, Ihre ersten Schritte mit einer SPS zu erleben. Das Programm arbeitet auf Simulationsbasis, und um das System zu vervollkommnen, ist es noch möglich, über eine Weiterentwicklung des I/O-Ports (aus dem Heft 4/87 der Zeitschrift ST-Computer) eine Verbindung zur Außenwelt zu schaffen. Somit können Sie Ihren ATARI ST für Steuerungszwecke einsetzen.

Alle folgenden SPS-Programmierbeispiele können Sie mit diesem Programm nachvollziehen! Falls Sie also das Programm „SPS-Emulator" der Sonderdiskette SD014 besitzen, so lesen Sie bitte die Bedienungsanleitung und starten Sie das Programm dann. Verzweigen Sie in den Editor und in den automatischen Eingabe-/Korrekturmodus, damit Sie programmierbereit sind!

Vorher aber noch eine Bemerkung, wie die Programmierung auf einer echten SPS erfolgt. Jedes Automatisierungsgerät besitzt eine Schnittstelle, an der ein Programmiergerät oder ein mit spezieller Software versehener Rechner angeschlossen werden kann. Bei den weitverbreiteten Siemens-STEP-5-Geräten handelt es sich hier um eine V24/TTY-Schnittstelle. Das Programmiergerät oder der angeschlossene Rechner haben einen eigenen Arbeitsspeicher, in dem das SPS-Programm (d.h. die Anweisungsliste, kurz: AWL) geschrieben oder ediert wird. Das AG hat zwei Betriebszustände, einen STOP- und einen RUN-Betrieb. Der STOP-Betrieb dient dazu, mit dem Programmiergerät (kurz: PG) zu kommunizieren. Will man die aktuelle AWL, die im Arbeitsspeicher des AG liegt, verändern, muß das AG im STOP-Betrieb sein. Durch einen Tastaturbefehl greift das PG auf den Speicher des AG zu und transferiert den Speicherinhalt in den Arbeitsspeicher des PG. Hier kann die AWL ediert und, falls gewünscht, wieder über einen Tastaturbefehl in den Arbeitsspeicher des AG gebracht werden. Hierbei wird die alte AWL überschrieben. Wenn man nun das neue SPS-Programm testen will, schaltet man das AG auf RUN. Der Prozessor arbeitet die AWL mehrere tausend Mal pro Sekunde seriell ab und setzt nach jedem AWL-Durchlauf die Ausgänge in Abhängigkeit von den Verknüpfungsergebnissen. Bevor man auf einer SPS eine AWL zu einer Steuerung erstellen kann, braucht man einen kleinen Überblick über die Grundbefehle und die Syntax.

Die Grundbausteine der SPS-Programmiersprache

Eingänge: Bezeichnung für die Anschlußstellen der Eingangsbaugruppe. An den Eingängen werden die Steuergeber angeschlossen. Ein Eingang kann auf sein logisches Spannungspotential, HIGH = Eingangsspannung (24V) = „1" / LOW = 0 V = „0", hin abgefragt werden.

Ausgänge: Bezeichnung für die Anschlußstellen der Ausgangsbaugruppe. An den Ausgängen werden die Stellglieder angeschlossen. Ein Ausgang kann auf HIGH = „1" oder LOW = „0" liegen. Ist ein Ausgang auf HIGH, wird der Steuerstromkreis zu dem an diesem Ausgang angeschlossenen Stellglied geschlossen, womit dieses Stellglied aktiviert wird.

Merker: Ein Merker ist eine einzelne Speicherzelle, die den logischen Zustand „1" oder „0" haben kann. Merker werden in der SPS als Hilfsvariablen zur Speicherung von Verknüpfungsergebnissen oder zu Verriegelungen verwendet.

Zähler: Ein Zähler kann in der SPS mit einem bestimmten Wert geladen werden und dann abhängig von einem Verknüpfungsergebnis auf- oder abwärtszählen. Zähler finden Anwendung, um Mengen, Zyklendurchläufe oder Stückzahlen zu erfassen. Ein Zähler kann auf seinen Zählerstand und seinen logischen Zustand hin abgefragt werden. Ein Zähler, der mit einem Zählwert geladen wurde, liefert bei einer Abfrage seines logischen Zustandes den Wert „1". Wenn ein Zähler den Wert 0 erreicht, so ist sein logischer Zustand „0". Ein Zähler kann maximal mit dem Zählerwert 999 geladen werden.

Timer: Ein Timer ist ein programmierbares Zeitglied, das man mit einem bestimmten Zeitwert laden kann. Ein Timer läßt sich auf fünf verschiedene Zeitablaufbedingungen voreinstellen:

- einschaltverzögert
- abfallverzögert
- Impuls
- speichernder Impuls
- speichernd einschaltverzögert

Ist die Zeit eines gestarteten Timers abgelaufen, wird bei seiner Abfrage der Wert

„1“ ausgegeben. Einen Timer kann man auf logischen Zustand und nach abgelaufener Zeit abfragen. Er ist von 10 Millisekunden bis 9990 Sekunden zu laden.

Eingänge, Ausgänge und Merker bezeichnet man auf den Siemensanlagen mit einem 8-Bit-System:

Eingänge:
Ex.0, Ex.1, Ex.2, ..., Ex.7, E(x+1).0, E(x+1).1
...

Ausgänge:
Ay.0, Ay.1, Ay.2, ..., Ay.7, A(y+1).0, A(y+1).1
...

Merker :
M0.0, M0.1, M0.2, ..., M0.7, M1.0, M1.1 ...

x und y sind Zuordnungszahlen, die beschreiben, wie die Reihenfolge der auf dem Systembus nacheinander aufgesteckten Eingangs- bzw. Ausgangsmodule ist. Timer und Zähler werden dezimal durchnumeriert:

Zähler:
Z 01, Z 02, Z 03, Z 04, Z 05 ... Z 16

Timer:
T 01, T 02, T 03, T 04, T 05 ... T 16

Falls Sie den SPS Emulator für Ihren ATARI besitzen, werden Sie feststellen, daß hier auch die Ein-/Ausgänge und Merker dezimal durchnumeriert sind (01 - 20). In der neuesten Version (größer 4.7) ist es selbstverständlich möglich, auf die Adressierung der STEP-5-Geräte umzuschalten, und es sind optional 40 Eingänge und Merker möglich!

Die Grundbefehle

Ein SPS-Programm macht eigentlich nichts anderes, als Spannungszustände vom Eingangs- oder Ausgangsbaustein einzulesen und laut der AWL logisch zu verknüpfen. Doch nun sollen Ihnen die vier ersten Befehle zur Programmierung vorgestellt werden:

Der Begriff „Eingänge“ steht hier für alle auf einen logischen Zustand abfragbaren Bausteine wie Eingänge, Ausgänge, Merker, Zähler, Timer usw.

Die UND-Verknüpfung:
Eine UND-Verknüpfung fragt einen oder mehrere Eingänge auf den logischen Spannungspegel HIGH = 1 ab. Sind alle abgefragten Eingänge auf HIGH, wird das Verknüpfungsergebnis auf HIGH wechseln, ansonsten auf LOW.

Die Syntax des UND-Befehles:
U E 0.0 -> Eingang 0.0 auf „1“ abfragen
U M 0.1 -> Merker 0.1 auf „1“ abfragen
U A 2.1 -> Ausgang 2.1 auf „1“ abfragen
U Z 01 -> Zähler 01 auf „1“ abfragen
U T 05 -> Timer 05 auf „1“ abfragen

Die ODER-Verknüpfung:
Die ODER-Verknüpfung fragt einen oder mehrere Eingänge auf den logischen Spannungspegel HIGH = „1“ ab, und falls einer der abgefragten Eingänge auf HIGH liegt, wird das Verknüpfungsergebnis auf HIGH wechseln. Nur dann, wenn alle abgefragten Eingänge LOW sind, ist das Verknüpfungsergebnis auch LOW.

O E 0.0 -> Eingang 0.0 auf „1“ abfragen
O M 0.1 -> Merker 0.1 auf „1“ abfragen
O A 2.1 -> Ausgang 2.1 auf „1“ abfragen
O Z 01 -> Zähler 01 auf „1“ abfragen
O T 05 -> Timer 05 auf „1“ abfragen

Die NICHT-Verknüpfung:
Die NICHT-Verknüpfung arbeitet als Negation und kann aus einer UND- eine NAND-Abfrage und aus einer ODER- eine NOR-Abfrage machen. Somit lassen sich einzelne Ein-/Ausgänge negieren und erneut verknüpfen.

UN E 0.0 -> Eingang 0.0 auf „0“ abfragen
UN M 0.1 -> Merker 0.1 auf „0“ abfragen
UN A 2.1 -> Ausgang 2.1 auf „0“ abfragen
ON Z 01 -> Zähler 01 auf „0“ abfragen
ON T 05 -> Timer 05 auf „0“ abfragen

Der GLEICH-Befehl:
Dieser Befehl wird nach einer logischen Verknüpfung gesetzt und nur ausgeführt, solange das Verknüpfungsergebnis „1“ ist. Mit diesem Befehl kann man Ausgänge und Merker, abhängig von der Erfüllung der vorangegangenen Verknüpfung, solange auf „1“ halten, bis das Verknüpfungsergebnis wieder „0“ ist.

U E 0.0 -> Eingang 0. 0 auf „1“ abfragen
U E 0.1 -> Eingang 0. 1 auf „1“ abfragen
O E 0.2 -> Eingang 0. 2 auf „1“ abfragen
= A 0.0 -> wenn Bedingung erfüllt,
Ausgang 0.0 auf „1“ setzen

Ausgang A 0.0 ist aktiv, wenn Eingang E 0.0 und Eingang E 0.1 HIGH sind, oder wenn Eingang E 0.2 HIGH ist. Man kann auch schreiben:

A 0.0 = 1, wenn gilt (E 0.0 = 1 und E 0.1 = 1)
oder E 0.2 = 1

Beispiele:
Ab jetzt wird davon ausgegangen, daß Sie das Emulatorprogramm für den ATARI besitzen und die Programmierbeispiele simultan zum Text eingeben und ausprobieren.

Was sind Öffner und Schließer?

Es gibt zwei verschiedene Ausführungen eines Schaltkontaktes, den Öffner und den Schließer. In einem Schaltplan werden alle Kontakte, falls nicht extra markiert, in Ruhestellung gezeichnet. Wenn man sich nun einen Schaltplan anschaut, fällt einem auf, daß manche Kontakte geschlossen und manche geöffnet gezeichnet sind (siehe Bild 3). Einen Kontakt, der in Ruhestellung keine Spannung und bei Betätigung Spannung durchschaltet, nennt man Schließer, denn er schließt sich bei Betätigung. Ein Öffner schließt in der Ruhestellung den Stromkreis und öffnet diesen bei Betätigung.

Was ist Drahtbruchsicherung?

Sehr (lebens)wichtige Steuerkontakte werden mit einem Öffner realisiert und liefern bei Nichtbetätigung ständig ein HIGH-Potential an die Eingangsbuchse. Wird nun der Stromkreis durch den Schalter oder einen Defekt am Schalter/Leitung unterbrochen, so geht die Anlage in den Störbetrieb. Dieses System findet auch bei Alarmanlagen in den Sicherungsschleifen Verwendung.

Programmierbeispiel:
Eine Signalleuchte soll dann leuchten, wenn der Taster S1 (ein Schließer) betätigt und der Taster S2 (ein Öffner) nicht betätigt ist (siehe Bild 3). Die erste Darstellung zeigt die Lösung der Steuerung in der VPS. Die Abbildung daneben zeigt, wie die Geber und das Stellglied an der SPS angeschlossen werden und welche Ein- bzw. Ausgänge den Gebern und dem Stellglied zugeordnet wurden. Es folgt die Schilderung des Wegs der Umsetzung in die AWL. Zuerst stellt man eine Funktionstabelle auf, in der in den vorderen Spalten alle Geber und in den hinteren alle Stellglieder eingetragen sind. Man benötigt so viele Zeilen, wie es mögliche verschiedene Zustände der einzelnen Geber gibt. In unserem Fall sind es zwei Geber und somit 2^2 mögliche Zustände. Zwecks der Übersicht trägt man die einzelnen Zustandsmöglichkeiten im Binär-System ein, wobei eine „1“ für Spannung an der Klemme des AG und „0“ für keine Spannung steht: Bei dieser Steuerung also 0-0 / 0-1 /

Take 20

Hard- und Software

Take 20 - Tip des Monats: 20

Mini Drives

Die schnellen SCSI-Platten im Mini-Gehäuse

Mini Drive 80MB, 24ms **nur DM 1598,- ***
Klein, dezent und leistungsstark: Eickmann Mini Drives - mit hardwaremäßigem Schreibschutz - wurden gezielt auf die Platzeinsparung und freie Plazierungsmöglichkeit hin konzipiert. Das Gehäuse ist im Design der Mega-Serie gehalten, aber kaum halb so groß! Mini Drives werden serienmäßig mit HD-Plus 5.02 (Cache, Paßwort, Schreibschutz, Bootwahl, etc.), Fast Filemover, HDU und Optimizer geliefert.

Date Access V2.0 -Der Terminplaner- jetzt in 3 Versionsarten! 1
DATE-Piccolo: Terminverwaltung mit allen Features. Termine können kopiert oder verschoben werden. Mit Tages-, Wochen- und Monatsübersicht. 59,- * DM.
DATE-ACCESS: wie oben, aber mit DATEHELP PRG zum Einbinden von Adressen. 79,- * DM.
DATE-JURIS: spezieller Planer für Juristen. Für Wiedervorlage- und Fristentermine. 99,- * DM.
DEMO zu Date Access 15,- * DM.

FolioTalk **DM 98,- *** 7
Interfaceprogramm und Verbindungskabel zur Datenübertragung zwischen Atari ST und Portfolio. Die Übertragungssoftware läuft als Accessory oder GEM-Anwendung und ermöglicht den einfachen und sicheren Datenaustausch vom ST auf den Portfolio und umgekehrt. Parallele Schnittstelle erforderlich.
Portfolio Komplettpaket **DM 598,- ***
Der kleinste PC der Welt, inkl. parallel Interface und FolioTalk (Schnittstelle zum ST).

Geben Sie Ihrem Calamus® die Fonts, die er braucht!

-------- je DM 39,- * --------
Plup Bobble
YAPPIE
Skript ROCA
-------- je DM 59,- * --------
AltBerlin bold
Peking reg Caslo
ROMA REG
CARDPLAY
-------- je DM 79,- * --------
AKTIVA Rund
GEODET JILLY
Boedet INFRA

Einfach mitnehmen: 2

Stacy LST 2/20 **4798,- ***
Portabler ST mit 2MB RAM, 20MB Festplatte und HD-Plus 5.0 im Paket.

Doppelte Kapazität:

Stacy LST 4/40 **6598,- ***
Portabler ST mit 4MB RAM, 40MB Platte und HD-Plus 5.0 im Paket.

Lern ST plus **DM 59,- *** 8
Maskenorientiertes Universal-Lernprogramm für Sprachen, Geschichte und Faktenwissen aller Art. Komfortabel und leistungsfähig.
Zusatzdisketten für Lern ST plus, je DM 20,- *

Englisch	(3000 Vokabeln)
Englisch 2	(3000 weitere Vokabeln u. Wendungen)
Französisch	(4500 Vokabeln und Wendungen)
Spanisch	(4500 Vokabeln und Wendungen)
Italienisch	(ca. 4000 Vokabeln und Wendungen)
Latein	(ca. 3000 Vokabeln und Wendungen)
Geschichte	(ca. 300 wichtige Ereignisse)
Führerschein	(ca. 400 Multiple-Choice-Fragen)

6
Syntex
die OCR
NEU 199,- * DM

STransPlus **DM 99,- *** 8
Elektronisches Wörterbuch mit 20.000 englischen Vokabeln. Per ACC-Funktion auch aus Textprogrammen bequem erreichbar. Die Integrierte Übersetzungshilfe liest ASCII-Text ein und zeigt gefundene Übersetzungen eines Wortes an. Einfache "Roh"-Übersetzungen (ohne grammatikalische, syntaktische oder inhaltliche Überprüfung) erstellt STransPlus auf Wunsch automatisch. Mit Vokabeltraining

AT-SPEED

So zuverlässig wie der PC-Speed und flink wie ein AT.

Emulieren Sie mal!

MegaPaint II **DM 199,- ***
Der Grafik-Profi
Soundmaschine **DM 99,- ***
Soundmerlin **DM 149,- ***
Graphstar **DM 49,- ***
DMA-Timer **DM 98,- ***
DMA-Kabel mit Festplatten-Einschaltverzögerung: Rechner, Platte und Peripheriegeräte können gleichzeitig - z.B. mittels Steckerleiste - angeschaltet werden, der Bootvorgang wird vom Timer koordiniert.
3,6m Tastaturkabel **DM 49,- ***
Spiralkabel für die Tastatur des MEGA ST: Damit Sie auch vom Sofa aus bequem arbeiten können.
EM 124 s/w Multisync Monitor **DM 498,- ***
Stellt Farbprogramme in Graustufen dar. Auch als Umbau Ihres SM124 Monitors lieferbar.
ST-Uhr 2 für 520/1040 ST **DM 79,- ***
Auf Soundchip steckbar, daher sehr einfach einzubauen.

Btx/Vtx Manager **DM 389,- *** 7
Softwaredecoder für BTX und Videotext, mit Interface an Postbox
Btx/Vtx Manager **DM 289,- ***
Wie oben, jedoch ohne Interface. Zum Anschluß an Akustikkoppler/Hayes-Modem (FTZ-Zulassung beantragt).
Entdecken Sie neue Informationsquellen: Sie finden uns - mit weiteren attraktiven Soft- und Hardware-Angeboten - auch im BTX-Anbieterverzeichnis der Deutschen Bundespost.

Their Finest Hour **DM 99,- *** 9
Außergewöhnlich detaillierte und realistische Simulation der Luftschlacht um England. Acht Maschinen vom Typ Spitfire, Hawker Hurricane, Me 109, Me 110, Ju 87 Stuka, Junkers 88, Dornier 17 und Heinkel 111 sind originalgetreu nachempfunden und perfekt simuliert. Als Pilot, Bombardier oder Bordschütze fliegen Sie dutzende von historischen Missionen oder stellen neue Einsätze im Mission-Designer zusammen. LUCASFILM setzt wieder Maßstäbe.

Hersteller- und Händleranfragen erwünscht ...
Wir suchen kompetente Kollegen, die sich an gemeinsamen Anzeigen in ST-Computer, ST-Magazin und TOS-Magazin beteiligen möchten. Bitte sprechen Sie mit einem der nebenstehenden Anbieter.

Autoren gesucht ...
Wir suchen attraktive Soft- und Hardwareprodukte zur Vorstellung auf diesen Seiten.

Skizzenbücher von Volkmar Hoppe entdeckt! 11

Geh! Hirn

Wer mit Text- und DTP-Programmen arbeitet, braucht hochwertige und aussagekräftige Grafiken zur copyright-freien Verwendung: Der Künstler Volkmar Hoppe hat seiner Phantasie freien Lauf gelassen und unzählige reale wie irreale Assoziationen in seine Arbeit einfließen lassen. „Bildhafte Wortspiele, satirische Anmerkungen, zynische Reflexionen, Sinnvolles, aber auch Spielerisches und Witziges stellen das künstlerische Ergebnis dar, das mit geübter Feder zu Papier bzw. auf die Diskette gelangt ist. Das komplette Paket umfaßt 17 Disketten.

Brainstorm 4 by Hoppe **DM 189,- ***

120 Seiten DIN A4 auf 17 Disketten
Alle Bild-Dateien im GEM Image-Format: In Calamus und anderen DTP-Programmen direkt verwendbar. Konvertierungsprogramm erzeugt ST-Standard Bildformate für Signum, STAD etc. 2MB Speicherplatz erforderlich

Architektur & Design **DM 59,- *** 10

Zwei Disketten mit mehr als 400 Raster- und Vektorgrafiken im (PAC- und GEM-Format). Menschen, Rahmen, Bäume, Pflanzen, Schiffe, Autos, Pfeile, Sterne etc...



Love & Fun 13

zeigt, welcher Partner zu Ihnen paßt: In der Liebe, beim Sport und am Arbeitsplatz. Kern der Profi-Version ist die schnelle Adressverwaltung, deren gesamter Datenbestand zur Partnersuche herangezogen werden kann. Mächtige Analysemöglichkeiten, flexible Suchkriterien, einfache Bedienung und die elegante Benutzeroberfläche zeichnen Love&Fun aus.

Love&Fun **DM 59,- ***

Manche wollen's einfach wissen ...

11

Calamus	**DM 798,- ***
Outline Art	**DM 398,- ***
Reprostudio ST	**DM 498,- ***
Retouche	**DM 399,- ***
PKS-Write	**DM 198,- ***
Calamus Fonteditor	**DM 198,- ***
DIDOT Fonteditor	**DM 198,- ***
Layout-Paket für Calamus	**DM 79,- ***

Gestaltungshilfe, Pass- und Schnittmarken in DIN Standartformaten A5, A4 und A3, hoch u. quer. Mit Anleitung (auf Wunsch in Englisch).

mChem **DM 99,- *** 16

Das Makropaket verwandelt Signum 2 in ein professionelles Textsystem für Chemiker. Geniale Lösungen sind oft verblüffend einfach: Wo früher mit Schere und Klebstoff gebastelt wurde, arbeiten heute 270 Makros. Moleküldarstellungen werden hier ganz einfach im Baukastenprinzip erzeugt, verändert und mit gleichbleibender Qualität reproduziert. Integration von Text und Grafik in einem Programm und einem Arbeitsgang. Another milestone in publishing ...

ARTWORKS-BUSINESS **DM 398,- *** 11

Drei Disketten mit Gestaltungshilfen für Calamus: Briefbögen, Visitenkarten, Aufkleber, Formulare, Passermarken und vieles mehr. Alles ist fertig gestaltet und im Handbuch mit vielen nützlichen Informationen zu Layout, Gestaltung und Druckvorlagenerstellung abgebildet, und das auch noch mehrfarbig. ARTWORKS-BUSINESS erhalten Sie im stabilen Schuber.

Architektur & Design **DM 59,- ***

Über 400 Grafiken im Vektor und Pixelformat.

EX 40W60 **DM 2998,- *** 20

Festplatte und Wechselplatte in einem: Keine zusätzlichen Kabel, keine unansehnlichen Gehäuse-Stapel, durchgeschleifter DMA-Bus, autobootfähig, von außen adressierbar, einfachstes Back-Up von Platte auf auswechselbare Medien. Die zusätzliche Festplatte ist natürlich leise und schnell. Handbuch, Festplatten-Utilities HDPlus, HDCache, HDU, Optimizer und Foliotalk sind im Lieferumfang enthalten. Auch als Einbaukit für die Wechselplatte lieferbar. Technology at it's finest ...

LogiMouse Pilot **DM 99,- *** 11

Präzise und zuverlässig. Ergonomisches, der Hand angepaßtes Design für ermüdungsfreies Arbeiten selbst bei langem Einsatz. Ausgezeichnet mit dem Preis 'Industrie-Form Hannover'. Zwei (2) Jahre Garantie!

Mausmatte **DM 15,- ***

Marconi Trackerball **DM 198,- ***

100%tiger Mausersatz mit hoher Lebensdauer. Schnelles und präzises Ansteuern, sauberes Ziehen und Bewegen, keine "Wackler" beim Klicken. Der DTP- und CAD-Profi.

Imagine ... 20

... 4 MB RAM, insgesamt 444 MB auf 3 Festplatten, dazu 2 (oder 3?) Floppy-Laufwerke, eingebaute Emulatoren und Großmonitortreiber. Stellen Sie sich weiter vor, wir betreiben das Ganze über einen 68020er Prozessor 400% schneller als auf dem MEGA ST und vor Ihnen befindet sich nur die Tastatur und der Monitor! Die Lösung ist unterm Tisch: Tower of Power. Sie denken drüber nach, wir bauen den Tower dann nach Ihren persönlichen Anforderungen.

Printing Press **DM 99,- *** 11

Profi-Version und Nachfolger des beliebten PD-Programms, mit dem Sie viele alltägliche Druckaufträge abwickeln können. Briefköpfe, Grußkarten, Diskettenlabels, Etiketten, Bilder und Bildausschnitte in Postergröße, Schriftbanner, Kuchen- und Balkendiagramme werden mit PP gestaltet und in beliebiger Stückzahl gedruckt. PP verarbeitet GEM-Fonts, Signum-Fonts, alle gängigen Bildformate, sogar Adressdaten aus 1st Adress zur Serienbrieferstellung.

20

5.25" Laufwerk **DM 348,- ***
720 KB anschlußfertig, 40/80 Tracks

3.5" Laufw.	720KB anschlußfertig	**DM 248,- ***
EX 30L	Festplatte	**DM 1298,- ***
EX 60L	Festplatte	**DM 1798,- ***
EX 40W	Wechselpl. & Medium	**DM 1998,- ***
EX 40W80	Wechsel-/Festplatte	**DM 3298,- ***

80MB Festplatte 24ms. plus 40MB Wechselplatte

EX Fest- und Wechselplatten werden mit HD-Plus 5.02 (Cache, Paßwort, Schreibschutz, Bootwahl, etc.), Fast Filemover, HDU und Optimizer geliefert.

Die vorgestellten Produkte erhalten Sie bei einem der folgenden Anbieter:

HD Computertechnik
Pankstr. 42
1000 Berlin 65
030 / 4657028-29

T.U.M.-Soft&Hardware
Hauptstr. 67
2905 Edewecht
04405 / 6809

HESSE & HERWIG
Lesteweg 33
4790 Paderborn
05251 / 92231

H. Richter Distributor
Hagener Str. 65
5820 Gevelsberg
02332 / 2706 (Fax: 2703)

Eickmann Computer
In der Römerstadt 249
6000 Frankfurt 90
069 / 763409 (Fax: 7681971)

IDL Software
Lagerstraße 11
6100 Darmstadt 13
06151 / 58912 (Fax: 591050)

Duffner Computer
Habsburgerstr. 43
7800 Freiburg
0761 / 56433 (Fax: 551724)

T.S. Service
Szemere Hard & Software
Schleißheimerstr. 127, 8 Mü 40
089 / 3089408

Schöll Büroteam
Dominikanerplatz 5
8700 Würzburg
0931 / 308080 (Fax: 3080821)

Bossart-Soft
Sonnenhofstr. 25 / PF 5146
CH-6020 Emmenbrücke 3
041 / 458284

* Alle Preise sind unverbindlich empfohlene Verkaufspreise

Coupon

Ich bestelle

Name, Vorname

Straße

Plz. Ort STC 1/91

☐ Per Scheck Zuzüglich DM 5,- Versandkosten, unabhängig von der bestellten Stückzahl
☐ Per NN (Nur Inland, zuzügl. DM 8,- NN-Gebühr)

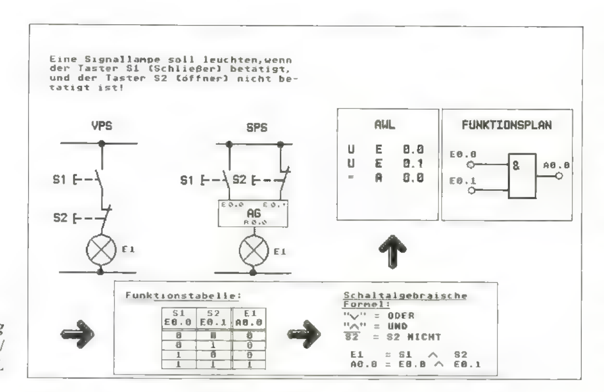


Bild 3: Umsetzung Stromlaufplan/ AWL

1-0 / 1-1. Dies sind alle Kombinationen, wie man die zwei Taster drücken kann und wie sich die Spannungspegel am Eingangsbaustein des AG ändern. Nun wertet man anhand der Verknüpfungsvorschrift aus der Aufgabenstellung aus, wann die Leuchte leuchten darf und wann nicht. Beginnen wir al-so in der ersten Zeile: Hier sind beide Spannungspegel auf „0", und somit sind der Schließer S1 nicht und der Öffner S2 betätigt. Dies entspricht nicht der Verknüpfungsvorschrift, also wird in diese Zeile unter dem Stellglied eine „0" eingetragen. In der nächsten Zeile sind S1 und S2 nicht betätigt also wird ebenfalls eine „0" eingetragen. Bei der folgenden Zeile sind S1 und S2 betätigt, somit wird ebenfalls eine „0" eingetragen. In der letzten Zeile trifft die Schalterstellung genau auf die Aufgabenstellung zu, somit ist eine „1" einzutragen.

In der Funktionstabelle existiert eine Zeile, die die Spannungspegel der Eingänge E 0.0 (S1) und E 0.1 (S2) angibt, damit der Ausgang A 0.0 (E1) Spannung führen darf. Mit dieser Zeile kann man die schaltalgebraische Formel der Steuerung aufstellen. In dieser Formelschreibweise werden drei Symbole verwendet, die eventuell einer Erklärung bedürfen: das Zeichen, das wie ein Dach (^) aussieht und nach UNTEN geöffnet ist, ist das Zeichen für „UND". Ist das Dach nach OBEN geöffnet, steht das Zeichen für „ODER". Ist eine Eingangskennzeichnung (hier ist es S2) überstrichen, steht dies für S2 „NICHT".

Nun können wir die Formel aufstellen! Ausgang A 0.0 soll HIGH sein, wenn E 0.0 = „1" UND wenn E 0.1 = „1". Wenn Sie die Formel auf Bild 3 betrachten, sehen Sie diese Bedingung mit den oben erklärten Zeichen aufgeschrieben. Diese Methode der Formelschreibweise ermöglicht es, sehr komplexe Verknüpfungen einfach aufzugliedern, und ein weiterer Vorteil ist, daß man aus dieser Formel die spätere AWL sehr leicht herauslesen kann. Die AWL zu dieser Steuerung steht auf der rechten Seite von Bild 3. Daneben wurde ein sogenannter Funktionsplan zur AWL erstellt. Mit dem Funktionsplan werden die Verknüpfungen der AWL mit Hilfe von logischen Gattern (UND, ODER, NICHT ...) dargestellt. Ein geübter SPS-Programmierer kann in beide Richtungen, also von der AWL zum FUP und umgekehrt, programmieren.

Der SPS-Emulator bietet Ihnen die Möglichkeit, zu einer sich im Arbeitsspeicher befindenden AWL den Funktionsplan zu erstellen und auf Wunsch auszudrucken.

Geben Sie die AWL mit Hilfe des Editors ein (falls Sie die Emulatorversion kleiner als 4.8 besitzen, müssen Sie für E 0.0 = E01 für E 0.1 = E 02 und für A 0.0 = A 01 eingeben), verlassen Sie den Editor und bringen den Emulator in RUN-Betrieb. Ändern Sie die Spannungszustände von E 01 und E 02 (Version kleiner als 4.8) und beobachten Sie die Reaktion des Ausganges A 01. Soeben haben Sie Ihre erste SPS-Steuerung eingegeben und getestet!

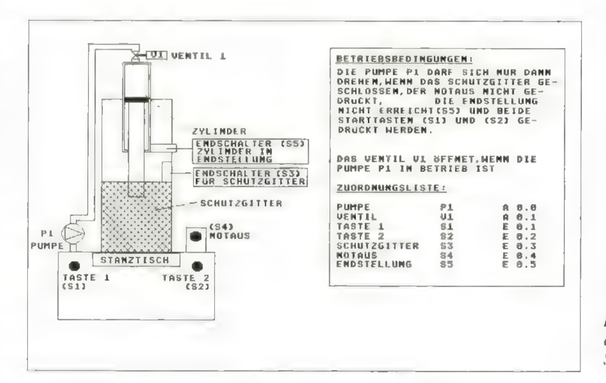


Bild 4: Steuerung einer Stanzmaschine

Dies war natürlich eine sehr simple Steuerung, bei der sich ein Einsatz einer SPS niemals gelohnt hätte. Deshalb ist das nächste Beispiel (Bilder 4 und 5) das technische Problem einer Stanzensteuerung. Die Stanze verfügt über die Hydraulikpumpe P1, die das Öl durch das Ventil V1 in den Zylinder preßt und diesen damit nach unten drückt. Um den Stanztisch ist ein Schutzgitter angebracht, das bei geöffnetem Zustand den Endschalter S3 betätigt. An dem Stanztisch ist der NOT-AUS-Schalter S4 installiert, und unter der Arbeitsplatte, in großem Abstand, sind die zwei Bedienknöpfe (S1+S2) des Stanzarbeiters, womit dieser den Stanzvorgang einleiten kann. Es wurden zwei Bedienknöpfe eingebaut, um sicherzustellen, daß der Arbeiter während des Stanzens beide Hände außerhalb des Gefahrenbereichs hat. Wenn der Kolben des Zylinders seinen Maximalweg zurückgelegt hat, betätigt er den Endschalter S5.

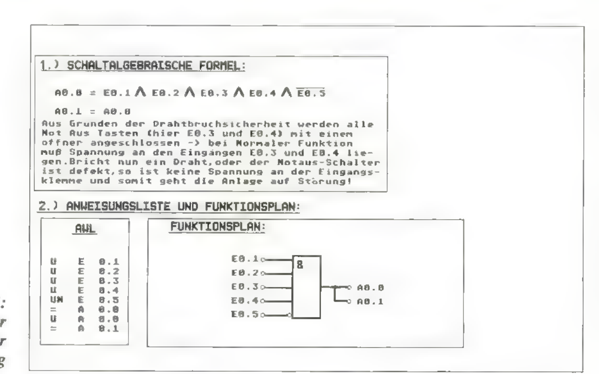


Bild 5: Erstellung der AWL zu dieser Steuerung

Nun die Bedingungen der Steuerung: Die Pumpe P1 darf nur aktiv sein, wenn das Schutzgitter geschlossen, sowie der NOT-AUS- und Endschalter S5 nicht betätigt und beide Starttasten S1 und S2 gedrückt sind. Das Ventil V1 muß sich öffen, solange die Pumpe in Betrieb ist. In Bild 4 sehen Sie eine schematische Skizze der Maschine. In Bild 5 wurden die schaltalgebraische Formel und schließlich die AWL erstellt. Die Schalter S3 und S4 wurden aus Gründen der Drahtbruchsicherheit als Öffner eingebaut und müssen deshalb auf den Zustand „1" abgefragt werden.

Geben Sie diese AWL ebenfalls ein und testen Sie auf Funktion. Um die Funktion der Drahtbruchsicherheit zu begreifen, fragen Sie doch mal S3 und S4 mit „UN" ab, gehen Sie in deh RUN-Betrieb und simulieren Sie einen Drahtbruch (S3 und S4 können niemals Spannung an die Eingangsklemme liefern), und nun versuchen Sie einmal, die Anlage in einem Notfall zu stoppen!

Mit dem „UND"-, „ODER"- und „NICHT"-Befehl können Sie nun schon sehr viele Steuerungsprobleme lösen und in die Programmiersprache der SPS umsetzen. Wie sehr es auf die Reihenfolge der Verknüpfungen in der AWL ankommt, sollen die zwei Abbildungen in der oberen Hälfte von Bild 6 zeigen: Erfolgt eine UND- vor einer ODER-Verknüpfung, wird das Verknüpfungsergebnis der UND- mit dem der ODER-Verknüpfung ODER-verknüpft. Folgt eine UND- auf eine ODER-Verknüpfung, so wird das Verknüpfungsergebnis der ODER-Verknüpfung mit dem der UND-Verknüpfung UND-verknüpft. Somit ergeben sich je nach Anordnung der einzelnen AWL-Zeilen verschiedene Funktionsabläufe, was bei der Programmerstellung beachtet und auch gezielt ausgenutzt wird.

Es gilt allerdings noch eine Besonderheit der Verknüpfung von UND- und ODER-Abfragen zu beachten (Bild 6 unten). Man spricht hier von UND-vor-ODER bzw. ODER-vor-UND: Durch die folgenden zwei neuen Befehle „U(„ und „O" bietet die Programmiersprache STEP-5 noch die Möglichkeit, Klammern zu setzen. Es gibt eine UND-Verknüpfungsklammer und eine ODER-Verknüpfungsklammer. Beim UND-Klammertyp werden ganze Verknüpfungsblöcke in Klammern eingeschlossen und mit der folgenden Klammer UND-verknüpft. Bei der ODER-Klammer werden die Verknüpfungsblöcke durch den „O"-Befehl getrennt und dann ODER-verknüpft (siehe Bild 6).

Die ODER-vor-UND-Verknüpfung:

```
U(          <- Klammerebene einleiten
O E 0.0     <- 1. Verknüpfungsblock
ON E 0.1
O E 0. 2
)           <- Ende der ersten Klammer
U(          <- Klammerebene einleiten
O E 0.3     <- 2. Verknüpfungsblock
U E 0.4
U E 0.5
)           <- Ende der zweiten Klammer
= A 0.0
```

Der Ausgang 0.0 wird nur dann auf HIGH gesetzt; wenn beide Klammern das Verknüpfungsergebnis „1" liefern, da sie ja UND-verknüpft werden.

Die UND-vor-ODER-Verknüpfung:

```
U E 0.0     <- 1. Verknüpfungsblock
UN E 0.1
U E 0.2
O
U E 0.3     <- 2. Verknüpfungsblock
U E 0.4
U E 0.5
= A 0.0
```

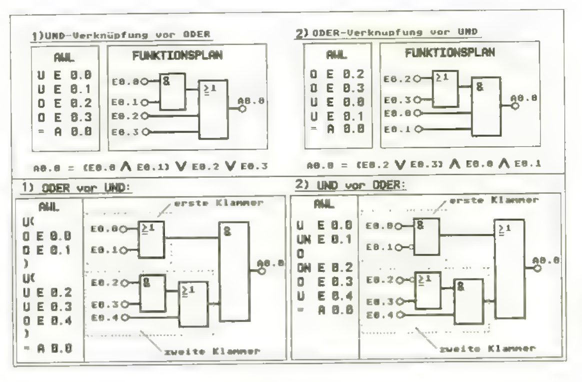


Bild 6: Logische Verknüpfung bei der SPS-Programmierung

Der Ausgang 0.0 wird nur dann auf HIGH gesetzt, wenn einer oder beide Verknüpfungsblöcke das Verknüpfungsergebnis „1" liefert.

Alle Versionen des SPS-Emulators, die eine größere Versionsnummer als 4.7 haben, verfügen über diese zwei Befehle.

Es ist jetzt an der Zeit, Ihnen die zwei nächsten und ebenfalls sehr wichtigen Befehle der SPS-Programmiersprache vorzustellen. Bei diesen beiden Befehlen handelt es sich um den „SETZE"- und den „RÜCKSETZE"-Befehl. Mit dem Setze-Befehl können Sie einen Ausgang, Merker dauerhaft und unabhängig von weiteren Verknüpfungsergebnissen, in den logischen Zustand „1" versetzen. Dieser Zustand wird beibehalten, bis er mit dem Rücksetze-Befehl wieder auf „0" gelegt wird. Diese programmierte Funktion ist mit dem J/K-Flip-Flop in der Elektronik vergleichbar. Die beiden Befehle sind sehr wichtig für die Registrierung von kurzen Tasterimpulsen, Verriegelungen usw. Die Funktion des SETZE- und RÜCKSETZE-Befehls soll Ihnen an folgendem Beispiel, einer Wendeschaltung mit zwei Sicherungsvorrichtungen (Schütze) demonstriert werden:

Syntax des Setze-Befehles:

```
S M 0.0 =   setze Merker 0.0 auf „1"
S A 0.1 =   setze Ausgang 0.1 auf „1"
```

Syntax des Rücksetze-Befehls:

```
R M 2.1 = rücksetze Merker 2.1 auf „0"
R A 0 2 = rücksetze Ausgang 0.2 auf „0"
```

Beispiel: (Wendeschaltung eines Motors) Ein Motor verfügt über zwei Drehrichtungen. Die Steuerung soll so angelegt werden, daß über den Taster S2 die rechte (Schütz an Ausgang 0.0) und über die Taste S3 in die linke Drehrichtung (Schütz an A 0.1) gewählt werden können. Der Taster S1 soll als NOT-AUS dienen und ist aus Gründen der Drahtbruchsicherheit als Öffner anzuschließen! Die Umschaltung von einer Drehrichtung in die andere soll vollautomatisch erfolgen, d. h. daß von einer Drehrichtung direkt in die andere geschaltet werden kann. Desweiteren ist ein gleichzeitiges Anziehen der zwei Motorschütze aus Kurzschlußsicherheitsgründen mittels einer Verriegelung zu verhindern.

Zuweisungsliste:

```
S1 (Öffner) NOT-AUS           =  E 0.0 (bzw. E01)
S2 (Schließer) Rechtslauf     =  E 0.1 (bzw. E02)
S3 (Schließer) Linkslauf      =  E 0.2 (bzw. E03)
Schütz Rechtslauf             =  A 0.0 (bzw. A01)
Schütz Linkslauf              =  A 0.1 (bzw. A02)
```

AWL:

```
U E 0.0     (NOT-AUS nicht betätigt?)
U E 0.1     (Taster für Rechtslauf betätigt?)
```

UN E 0.2	(Taster für Linkslauf nicht gedrückt?)
R A 0.1	(Schütz Linkslauf in Ruhestellung!)
S A 0.0	(Motor Rechtslauf!)
U E 0.0	(NOT-AUS nicht betätigt?)
U E 0.2	(Taster für Linkslauf betätigt?)
UN E 0.1	(Taster für Rechtslauf nicht gedrückt?)
R A 0.0	(Schütz Rechtslauf in Ruhestellung!)
S A 0.1	(Motor Linkslauf!)
UN E 0.0	(NOT-AUS gedrückt oder Drahtbruch?)
R A 0.0	(Motorstillstand durch Rücksetzen beider)
R A 0.1	(Motorschütze)

Geben Sie die AWL ein, bringen Sie den Emulator in den RUN-Betrieb, betätigen Sie E0.0/E01 und dann tasten Sie abwechselnd E0.1/E02 und E0.2/E03. Die Ausgänge A0.0/A01 und A0.1/A02 müssen nun abwechselnd HIGH sein. Wenn Sie E0.0/E01 unbetätigt lassen, so darf keiner der beiden Ausgänge auf „1" gesetzt werden. Nun wollen wir noch genauer die Verwendung von Merkern studieren. Wie oben schon gesagt wird ein Merker als Speichervariable, die den Zustand „1" oder „0" haben kann, verwendet. Merker verwendet man, um Verknüpfungsergebnisse zwischenzuspeichern. Das folgende Beispiel soll ein typisches Anwendungsbeispiel für einen Merker liefern: Die Signalleuchte E1 soll aufleuchten, wenn der Taster S1 betätigt wird und soll so lange leuchten, bis der Taster S2 betätigt wird. Bei diesem Beispiel handelt es sich um eine sogenannte Selbsthaltung. Diese Schaltungstechnik wird sehr oft zur Steuerung von Motoren verwendet (EIN/AUS-TASTER). Bei einer Selbsthaltung kann das Stellglied mit einem Tasterdruck eingeschaltet werden und es bleibt auch bei Nichtbetätigung des Starttasters in Funktion, man sagt es hält sich selbst. Erst durch Betätigung der Stoptaste fällt das Stellglied in die Ruhelage zurück.

Zuweisungsliste:

S1 (Schließer) EIN =	E 0.0 / E01
S2 (Schließer) AUS =	E 0.1 / E02
Anlagenzustand =	M 0.0 / M01
Signalleuchte E1 =	A 0.0 / A01

AWL:

U E 0.0	(Anlage EIN betätigt?)
S M 0.0	(Merker für Anlagenzustand auf „1"!)
U E 0.1	(Anlage AUS betätigt?)
R M 0.0	(Merker für Anlagenzustand auf „0"!)
U M 0.0	(Anlagenzustand „1"?)
= A 0.0	(Signalleuchte E1 leuchtet!)

Wenn Sie dieses Programm starten und E 0.0/E01 betätigen, geht Ausgang A 0.0/A01 auf „1" und bleibt in diesem Zustand, auch wenn Sie E 0.0/E01 nicht mehr betätigen, bis Sie E0.1/E02 betätigen. Die Krönung einer Selbsthaltungssteuerung ist eine Steuerung, bei der das Ein- und Ausschalten der Anlage über den gleichen Taster erfolgt. Der Ein- und Ausschaltbefehl wird alternierend, mit jeder Betätigung des Tasters, gegeben.

Zuweisungsliste:

S1 (Schließer) EIN/AUS =	E 0.0 / E01
Anlagenzustand =	M 0.0 / M01
Hilfsmerker =	M 0.1 / M02
Verriegelungsmerker =	M 0.2 / M03
Signalleuchte E1 =	A 0.0 / A01

AWL:

U E 0.0	(Taster gedrückt?)
UN M 0.0	(Merker für Anlagenzustand auf „0"?)
UN M 0.2	(Verriegelungsmerker auf „0"?)
S M 0.1	(Hilfsmerker auf „1"!)
U E 0.0	(Taster gedrückt?)
U M 0.0	(Merker für Anlagenzustand auf „1"?)
UN M 0.2	(Verriegelungsmerker auf „0"?)
R M 0.1	(Hilfsmerker auf „0"!)
U M 0.1	(Hilfsmerker auf „1"?)
S M 0.0	(Merker für Anlagenzustand auf „1"!)
UN M 0.1	(Hilfsmerker auf „0"?)
R M 0.0	(Merker für Anlagenzustand auf „0"!)
U E 0.0	(Taster betätigt?)
S M 0.2	(Verriegelung EIN!)
UN E 0.0	(Taster nicht betätigt?)
R M 0.2	(Verriegelung AUS!)
U M 0.0	(Merker für Anlagenzustand auf „1"?)
= A 0.0	(Signalleuchte EIN!)

Vielleicht stellt sich Ihnen nun die Frage, wozu die einzelnen Merker des Programmes dienen?

M 0.0: in diesem Merker wird der Betriebszustand der Anlage gespeichert „1" = Anlage EIN / „0" = Anlage AUS

M 0.1: dieser Hilfsmerker wird in allen Verknüpfungen anstatt M 0.0 eingesetzt, da sonst durch die serielle Abarbeitung der AWL, in den aufeinanderfolgenden Verknüpfungen von M0.0, dieser Merker nie den EIN-Zustand erreichen würde (ausprobieren! Ersetzen Sie M 0.1 durch M 0.0 und lassen sie das Programm ablaufen). M 0.2: ist ein Verriegelungsmerker der bewirkt, daß ein Betriebszustandswechsel nur dann erfolgt, wenn der Taster S1 zwischen einem EIN- bzw AUS-Befehl einmal unbetätigt war. Wenn Sie diesen Merker aus dem Programm entfernen, so wechselt die Steuerung (bei betätigtem EIN/AUS-Taster)bei jedem Programmzyklus den Betriebszustand.

Dieser Abschnitt sollte eine kleine Einführung in die Programmierung einer SPS sein und Ihnen zeigen, wie einfach und schnell ein Steuerungsproblem mittels der SPS gelöst werden kann. Mit diesen wenigen Befehlen der Programmiersprache Step 5 können Sie schon einige Steuerungsprobleme lösen. Weitere Befehle, die die Flexibilität einer SPS-Steuerung noch erhöhen sind Zähler und Timer, damit auch zeitkritische Abfragen in einer Steuerung beachtet werden können. Die Anbieter der SPS-Steuerungen erweitern das Angebot für ihre Automatisierungsgeräte ständig, um alle eventuellen Steuerungsprobleme lösen zu können. So werden zum Beispiel Analogbausteine angeboten, die einen Analogeingang mit wahlweise 0 ... 20 mA (Dreileitertechnik), 4 ... 20mA (Zweileitertechnik) oder 0 ... 30 V bieten und dieses Analogsignal digital umwandeln, damit die SPS den aktuellen Wert erfassen kann. Somit kann man zum Beispiel einen P-Transmitter, ein Gerät das in eine Druckluftleitung eingebaut wird und abhängig vom Druck einen Ausgangsstrom von 0 ... 20/4 ... 20 mA oder eine Ausgangsspannung von 0 ... 30 V liefert, direkt an der SPS anschließen. Die CPUs der SPS-AGs werden immer schneller und komfortabler, damit sie auch arithmetisch komplexe Berechnungen durchführen können. Der Trend läuft eindeutig zu komfortablen Mikroprozessorrechnern mit graphischer Benutzeroberfläche und entsprechender Perepherie, die ein Arbeiten mit diesen Systemen noch leichter und schneller machen.

Dieses Steuerungssystem, das den technischen Bereich von Ablaufsteuerungen schon revolutioniert hat und noch weiter revolutionieren wird, ist ein absolutes Muß für jeden, der in diesem Sektor in den nächsten Jahren bestehen will.

In der Ausbildung von vielen Berufen ist das Fach „SPS" schon zu einem Pflicht- und Prüfungsfach geworden und es ist jedem, der irgendwann einmal mit Steuerungsproblemen in Kontakt kommen kann, zu raten, daß er sich mit dem Thema SPS intensiv auseinandersetzt und sich immer auf dem neuesten Stand hält.

Thomas Doll

Einkaufsführer

Hier finden Sie Ihren Atari Fachhändler

1000 Berlin

Computer und mehr...



1000 Berlin 20
Breite Straße 50
Tel. (030) 33 32 008 • Fax 33 39 455



HD COMPUTER-TECHNIK oHG

PD SERVICE — Je Disk 5,- DM

Über 5000 PD-Disketten verfügbar!
z.B. ACS, AMIGA JUICE, AMIGALIBDISK, ANTARES, AUGE 4000, AUSTRIA, BAVARIA, BB, AMI, BRUNOSOFT, CACTUS, CHEMIE, CHIRON, ES PD, FAUG, FRANZ, GERMAN, GET IT, KIRSCHBAUM, MIDI PD, OASE, OLLIS GAMES, PANORAMA, PAXUM, PFALZ, PUDOMIX, R-H-S, RPD (=CAM), 6 2 6, SCHATZ, S-DREAMS, TAIFUN, TAURUS, TBAG, TIGER, TIME, TORNADOS, U.G.A., u. a.
und natürlich sämtliche KICKSTART-PD's!

- Speichererweiterungen
- Diskettenlaufwerke
- Festplatten & Turbokarten
- Anwendungssoftware
- Disketten Großhandel
- Fachliteratur & Zubehör
- Desktop Video/Publishing
- Reparatur Service

HD COMPUTERTECHNIK oHG
Pankstraße 61
1000 Berlin 65
Tel.:030/465 70 28

REPARATUR WERKSTATT
1000 Berlin 65 - Pankstr. 42

SERVICE STATIONEN
1/44, Lahnstr. 94, Tel:684 48 31
1/20, Schönwalder Str. 65, Tel:375 60 13

1000 Berlin

COMPUTERSYSTEME
Schlichting
...die etwas andere Computerei
COMPUTERSYSTEME, PLAYSOFTSTUDIO SCHLICHTING
COMPUTER - SOFTWARE - VERSAND GMBH & CO. KG
ATARI-FACHMARKT
MS-DOS FACHMARKT · NEC FACHHANDEL
Mönchstraße 8 · 1000 Berlin 20
030/786 10 96

DATAPLAY
Bundesallee 25 · 1000 Berlin 31
Telefon: 030/861 91 61

Computer und mehr...



1000 Berlin 47
Johannisthaler Chaussee 301
Tel. (030) 60 34 056 • Fax 60 24 058

COMPUTERSYSTEME
Schlichting
...die etwas andere Computerei
COMPUTERSYSTEME, PLAYSOFTSTUDIO SCHLICHTING
COMPUTER - SOFTWARE - VERSAND GMBH & CO. KG
ATARI-FACHMARKT
MS-DOS FACHMARKT · NEC FACHHANDEL
Wilh.Pieck-Straße 56 · Potsdam
030/786 10 96

1000 Berlin

ATARI
... wir machen Spitzentechnologie preiswert.
Vertragshändler
UNION ZEISS
Kurfürstendamm 57 • 1000 Berlin 15
Telefon 32 30 61

PLAYSOFT-STUDIO
Schlichting
...die etwas andere Spielerei
PLAYSOFTSTUDIO SCHLICHTING
COMPUTER - SOFTWARE - VERSAND GMBH
BERLINS NEUE DIMENSION FÜR COMPUTERSPIELE
Katzbachstraße 8 · 1000 Berlin 61
030/786 10 96

Steglitz Schloßstraße
030/79001-418
Ihre Tür zur Zukunft:
karstadt-
computer-center
hardware-software-problemlösungen

Computer und mehr...





1000 Berlin 48
Buckower Chaussee 100
Tel. (030) 72 16 004 • Fax 72 71 94

alpha computers g.m.b.h.
u. a. alphatronic, atari, commodore dai, epson, sord mit pips, nec
hard-/software nach maß — servicetechnik
Kurfürstendamm 121a, 1000 Berlin 31 (Halensee)
Telefon 030/891 1082

COMPUTERSYSTEME
Schlichting
...die etwas andere Computerei
COMPUTERSYSTEME, PLAYSOFTSTUDIO SCHLICHTING
COMPUTER - SOFTWARE - VERSAND GMBH & CO. KG
ATARI-FACHMARKT
MS-DOS FACHMARKT · NEC FACHHANDEL
Katzbachstraße 8 · 1000 Berlin 61
030/786 10 96

2000 Hamburg

Planen • Beraten • Realisieren



ATARI Fachhändler

Münsterstraße 9, D-2000 Hamburg 54
Telefon 040/56 19 09-0
Telefax 040/56 19 09-80

RADIX Bürotechnik

Heinrich-Barth-Straße 13
2000 Hamburg 13
Telefon (0 40) 44 16 95

NEU: Software Shop



DTP u. Werbeservice

Soft- und Hardwareversand

Viola Jaap
Berner Heerweg 512
2000 Hamburg 72
Telefon (0 40) 6 44 06 68

2000 Norderstedt



2120 Lüneburg

Sienknecht
Bürokommunikation
Beratung - Verkauf - Werkstatt
Heiligengeiststr. 20, 2120 Lüneburg
Tel. 04131/46122, Btx 40 24 22
Mo.-Fr. 9⁰⁰-18⁰⁰ und Sa. 9⁰⁰-13⁰⁰

2210 Itzehoe

Der Computerladen

Coriansberg 2 · 2210 Itzehoe
Telefon (0 48 21) 33 90/91

2300 Kiel



Computer

MCC Computer GmbH Holzkoppelweg 19a
2300 Kiel 1 TEL 0431/54381 FAX 541717

2400 Lübeck



2400 Lübeck-Moisling
August-Bebel-Straße 25-27
Tel. (04 51) 80 20 14 • Fax 80 44 08

2800 Bremen



Faulenstraße 48–52
2800 Bremen 1
Telefon (04 21) 17 05 77

2940 Wilhelmshaven

Radio Tiemann
ATARI-Systemfachhändler
Markstr. 52
2940 Wilhelmshaven
Telefon 0 44 21 - 2 61 45

3000 Hannover



DATALOGIC COMPUTERSYSTEME
ATARI ST-COMPUTER · HARDWARE · SOFTWARE
BERATUNG · SERVICE · VERKAUF
CALENBERGER STR 26
3000 HANNOVER 1
TEL 0511 - 32 64 89

COM DATA

Am Schiffgraben 19 3000 Hannover 1
Telefon 05 11 - 32 67 36

3000 Hannover



HD COMPUTER-TECHNIK oHG

HANNOVER

Public Domain Service
Hardware & Zubehör
Fachliteratur

Hildesheimer Str. 118
3000 Hannover 1
Tel.: 0511/809 44 84

3400 Göttingen

Büroeinrichtungs-Zentrum



3400 Göttingen-Weende
Wagenstieg 14 – Tel 0551/38 57 0



3400 Göttingen
Weender Straße 75
Tel. (05 51) 4 80 86 • Fax 5 76 64

3500 Kassel

Hermann Fischer GmbH
autorisierter ATARI-Fachhändler

Rudolf-Schwander-Str. 5-13
3500 Kassel
Telefon (05 61) 70 00 00

4000 Düsseldorf

HOCO
EDV ANLAGEN GMBH
Ellerstraße 155
4000 Düsseldorf 1
Telefon 02 11/78 52 13

Hard und Software
Werner Wohlfahrtstätter

Atari
Public Domain
Atari Spiele
Atari Anwender

Ladenlokal
Irenenstraße 76c
4000 Düsseldorf-Unterrath
Telefon (02 11) 42 98 76

4000 Düsseldorf

BERNSHAUS GmbH
Bürotechnik – Bürobedarf
Cäcilienstraße 2
4000 Düsseldorf 13 (Benrath)
Telefon 02 11 - 71 91 81



4010 Hilden

Beachten Sie unsere Anzeige in diesem Heft!



Computer · Drucker · Software · Bücher · Service
Gustav-Mahler-Straße 42-44
Tel. (0 21 03) 3 18 80 + 4 12 26

4150 Krefeld



Kommanditgesellschaft
Nordwall 96
D 4150 Krefeld 1
Tel 02151-772056
FAX 02151-770995
BTX 02151772056

Computer- Hard- & Software
Ein- & Mehrplatzsysteme
Komplettlösungen



DTP-CENTER
MATRIX *MatScreen*
ROLAND *Schnittplotter*
DATACOPY *Scanner*
Digitizer

C.O.P COMPUTER GMBH
Ennsstrasse 20
Postfach 507
4150 Krefeld 1
☎ (0 21 51) 54 20 91
Fax (0 21 51) 54 20 94

4150 Krefeld



4200 Oberhausen

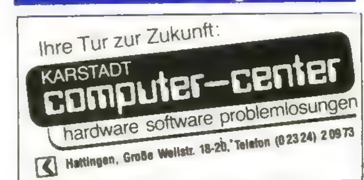
Redakteur • **Spectre 128**
dBMAN
für TOS, DOS, UNIX, MAC, AMIGA und andere Systeme
by COMPUTER MAI
dBMAN-Vertretung für NRW und BENELUX
ISYS-COMPUTER GbR
Tel.: 0208/655031 · Telefax 0208/650981
Max-Eyth-Straße 47 · 4200 Oberhausen 11



4300 Essen



4320 Hattingen



Bei uns werben bringt
GEWINN



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057

BUF

4400 Münster



4422 Ahaus

ATARI · Epson · Fujitsu
Molecular · NCR · Tandon · Schneider · Star
OCB

OCB-Computershop
Wallstraße 3
4422 Ahaus
Tel. 0 25 61/50 21

OCB-Hard- und Software
Wessumerstraße 49
4422 Ahaus
Tel. 0 25 61/50 21

4430 Steinfurt

CBS GmbH
COMPUTERSYSTEME
Tecklenburger Str. 27
4430 Steinfurt-Burgsteinfurt
☎ 02551/2555

4500 Osnabrück

Heinicke-Electronic
Meller Str. 43 • 4500 Osnabrück
Fax (0541) 58 66 14
Telefon (0541) 58 76 66
Wir liefern Micro-Computer seit 1978

4520 Melle

C B S GmbH
COMPUTERSYSTEME
4430 Steinfurt Tel. 02551/2555
Haferstraße 25 4520 Melle
Tel.: 05422/44788

4600 Dortmund

Elektronik
Computer
Fachliteratur
ATARI-System-Fachhändler
4600 Dortmund 1, Güntherstraße 75, Tel. (02 31) 57 22 84



4600 Dortmund

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Kampstraße 1 · 4600 Dortmund
Telefon (02 31) 5 43 91

cc Computer Studio GmbH



Elisabethstr. 5
4600 Dortmund 1
Tel. 0231 528184 Tx 822631 cccsd Fax 0231 528131

4650 Gelsenkirchen



Fest- und Wechselplatten (SCSI)
Großmonitore, Scanner, Software
ComputerSysteme und Anwendungen
Hüttenstraße 56 – 4650 Gelsenkirchen
Tel. (0209) ..3420 FAX (02 09) 21584

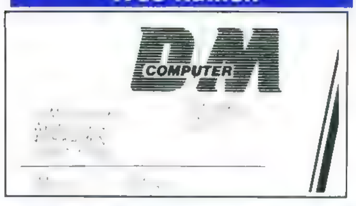
4650 Gelsenkirchen-Horst

MENTIS GmbH

Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel

Poststraße 15 · 4650 Gelsenkirchen-Horst
Telefon (02 09) 5 25 72

4708 Kamen



4800 Bielefeld

hardware
software
organisation
service
CSF

CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106 – 108
4800 Bielefeld 1
Tel. (05 21) 6 16 63

5090 Leverkusen

Rolf Rocke
Computer-Fachgeschäft
Auestraße 1
5090 Leverkusen 3
Telefon 0 21 71 / 26 24

5253 Lindlar



5300 Bonn

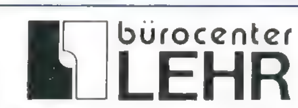


5430 Montabaur



5430 Montabaur
Industriestraße 20-24
Tel. (0 26 02) 34 67 • Fax 34 67

5500 Trier



Güterstraße 82 · 5500 Trier
☎ 06 51 / 20 97 10
Fordern Sie unsere Zubehör-Liste an!

5600 Wuppertal



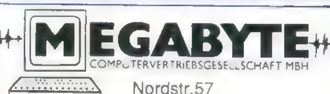
Kleiner Werth 18
5600 WUPPERTAL 2
Tel. (0202) 592064

5600 Wuppertal



5600 Wuppertal-Barmen 2
Unterdörnen 93
Tel. (02 02) 55 60 41 • Fax 57 15 42

5630 Remscheid



Nordstr.57
5630 REMSCHEID 1
Tel. (02191) 21033
Fax (02191) 21034

5650 Solingen

MegaTeam

Computer-Vertriebs-OHG
Kölbach – Finke

Hardware – Software – Zubehör – Service

Rathausstraße 1–3 · 5650 Solingen 1
Telefon (02 12) 4 58 88 · Fax (02 12) 4 73 99

5800 Hagen

ATARI
wir machen Spitzentechnologie preiswert.

Vertragshändler Axel Böckem
Computer + Textsysteme

Eilper Str. 60 (Eilpezentrum) · 5800 Hagen
Telefon (0 23 31) 7 34 90

6000 Frankfurt

WAIZENEGGER
Büroeinrichtungen

Kaiserstraße 41
6000 Frankfurt/Main
Tel. (0 69) 2 73 06 - 0

Eickmann Computer

***Der* Atari-System-Fachhändler !**

z.B.: Festplatten von 30 MB bis 110 MB für Atari ST und Mega ST, Zusatzaufrüstungen für Ihre Festplatten bis 110 MB, Umrüstung Ihres SM 124 in einen EM 124 Multisync für alle Auflösungen, Slotkit für PC 1...

besuchen Sie unser Fachgeschäft:
In der Römerstadt 249
6000 Frankfurt 90-Praunheim
Telefon (069) 763409

6100 Darmstadt

Heim
Büro- und Computermarkt
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon (0 61 51) 5 60 57

6108 Weiterstadt



6120 MICHELSTADT



ATARI ST / TT / PC PORTFOLIO
Desktop Publishing
Drucker Festplatten 19" Monitore Scanner
Software Speichererweiterungen Zubehör
Bürodrehstühle – Büroeinrichtungen
Tel. 06061/73601 FAX 06061/73602

6200 Wiesbaden





6236 Eschborn



6240 Königstein

KFC COMPUTERSYSTEME
Wiesenstraße 18
6240 Königstein
Tel. 0 61 74 - 30 33
Mail-Box 0 61 74 - 53 55

6250 Limburg



DTP-Center • Salzgasse 6 • 6250 Limburg
☎ 06431/5004-0 • FAX 06431/5004-10

6400 Fulda

Schneider ATARI Commodore
WEINRICH
BÜRO · ORGANISATION
Ronsbachstraße 32 · 6400 Fulda
Telefon (06 61) 4 92 - 0

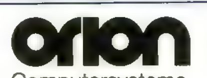
6457 Maintal



6500 Mainz - Hechtsheim



6520 Worms



Computersysteme
GmbH
6520 Worms · Friedrichstraße 22
Telefon 0 62 41 / 67 57 - 58

6630 Saarlouis



6700 Ludwigshafen

MKV Computermarkt
Bismarck-Zentrum
6700 Ludwigshafen
Telefon 06 21 - 52 55 96

6720 Speyer

THEILLE
Computersysteme
Gilgenstraße 4 · 6720 Speyer
Telefon (0 62 32) 772 16

Etzkorn Computer
ATARI *Desktop Publishing Center*
Vom Portfolio über ST bis zum TT alles lagermäßig und vorführbereit auf 10 Anlagen!
3K • DMC • Eizo • NEC • Protar • Spectre • Vortex
6720 Speyer • Auestr. 20
Tel. 06232 / 32435 • Fax 41398
Mo - Fr. 9 - 12 u. 15 - 18.30, Sa. 9 - 14 Uhr

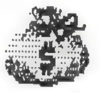
6800 Mannheim



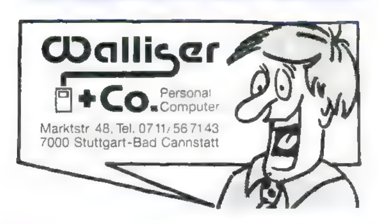
Computersysteme + Textsysteme
6800 Mannheim 24
Casterfeldstraße 74-76
☎ (06 21) 85 00 40 · Teletex 6 211 912

6900 Heidelberg

JACOM FAMILA-CENTER
Hardware · Software
Schulung · Service
Hertzstraße 1 · 6900 Heidelberg 1
Telefon (0 62 21) 30 00 44-5

Bei uns werben bringt
GEWINN

Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057
BUF

7000 Stuttgart



7100 Heilbronn



Computer-Welt
Am Wollhaus 6
7100 Heilbronn
Tel. 0 71 31 - 6 84 01 - 02

7150 Backnang



Bei uns werben bringt
GEWINN
Sprechen Sie mit uns.
Heim Verlag ☎ 0 61 51 / 5 60 57
BUF

7312 Kirchheim/Teck

Comp & Phone
Computerfachhandel
Alleenstraße 66
7312 Kirchheim/Teck
Tel.: 07021/3949 - Fax: 07021/53933

7475 Meßstetten

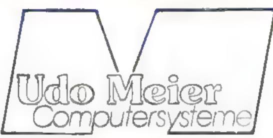
Ihr ATARI-Systemhändler im Zollern-Alb-Kreis
HEIM + PC-COMPUTERMARKT
HARDWARE · SOFTWARE · LITERATUR
SCHEURER
ATARI COMMODORE CUMANA DATA BECKER
MULTITECH RITEMAN SCHNEIDER THOMSON
7475 Meßstetten 1 · Hauptstraße 10 07431 / 6 12 80

7500 Karlsruhe

MKV GMBH
Kriegsstraße 77
7500 Karlsruhe
Telefon (0721) 8 46 13



7700 Singen



Ringstraße 4
Telefon (07731) 68222

7750 Konstanz

neu: ATARI Desk Top Publishing-Center
computer - fachgeschäft
rösler
Rheingutstr. 1 · ☎ 0 75 31-2 10 32

7890 Waldshut-Tiengen

hettler-data
service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 0 77 51 / 30 94

7910 Neu - Ulm






Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057
BUF

7918 Illertissen

bio-tech gmbh
technische Informationssysteme
Computerladen
Marktplatz 13
7918 Illertissen
0 73 03 / 50 45

7930 Ehingen



8000 München

T.S. Service
Szemere Hard & Software
Schleißheimerstr. 220
D-8000 München 40

Tel. 089/3089408 Fax 089/3085636 ATARI
Festplatten / Rechner / Zubehör / Service
Bürozeiten: Mo. bis Fr. 9.00-12.00 13.00-18.30 Samstag 9.00-14.00

8000 München

ludwig

COMPUTER + BÜROTECHNIK
COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST

INGOLSTADTER STRASSE 62L
EURO INDUSTRIE-PARK • 8000 MÜNCHEN 45
TELEFON 089 / 31 81 95-0 • TELETEX 89 83 41

City Studio
Rindermarkt 6, 8000 München 2
Tel. 089 / 2 60 98 01

schulz computer

Schillerstraße 22
8000 München 2
Telefon (0 89) 59 73 39

Beratung · Verkauf · Kundendienst

Bei uns werben bringt

GEWINN



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057 BUF

8032 Gräfelfing



Pasinger Straße 94 8032 Gräfelfing
Telefon 089 - 8548823
Fax 089 - 8541764

8150 Holzkirchen



8200 Rosenheim



COMPUTER + BÜROTECHNIK
Kufsteiner Str. 11, 8200 Rosenheim
Tel. (08031) 38 00 30
Fax (08031) 1 53 34

8390 Passau

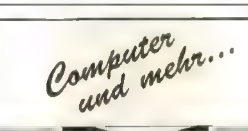


8400 Regensburg

WITTICH
COMPUTER GMBH
Luitpoldstr. 2
8400 Regensburg



Tel. (0941) 562530
Fax (0941) 562510





im ELEKTROLAND REGENSBURG
Dr.-Gessler-Straße 8
Tel. (09 41) 9 50 85 · Fax 94 79 61

8423 Abensberg

WITTICH
COMPUTER GMBH
Tulpenstr. 16
8423 Abensberg
☎ 0 94 43 / 4 53

8440 Straubing





Erfolgreich werben

Sprechen Sie mit uns.
Heim-Verlag ☎ (0 61 51) 5 60 57 BUF

8500 Nürnberg





8520 Erlangen

Computerservice Decker
Meisenweg 29 - 8520 Erlangen
Telefon 0 91 31 / 4 20 76



8600 Bamberg

Das Haus der großen Auswahl



8600 Bamberg · Am Kranen 12a
Tel. 09 51 / 2 78 08 · Fax 09 51 / 2 78 00



8700 Würzburg

SCHOLL
BÜROTEAM
Hardware · Software
Service · Schulung
computer center
am Dominikanerplatz
Ruf (09 31) 3 08 08 - 0

8720 Schweinfurt

Uhlenhuth GmbH
Computer + Unterhaltungselektronik
Albrecht-Dürer-Platz 2
8720 Schweinfurt
Telefon 0 97 21 / 65 21 54

8900 Augsburg

Adolf & Schmoll
Computer
Unser Plus: Beratung u. Service
Schwalbenstr. 1 · 8900 Augsburg-Pfersee
Telefon (08 21) 52 85 33 oder 52 80 87
Computer Vertriebs- und Software GmbH



8906 Gersthofen



8940 Memmingen



8960 Kempten



O-58oo Gotha



Bei uns werben bringt

GEWINN

Sprechen Sie mit uns.
Heim Verlag ☎ 0 61 51 / 5 60 57

BUF

O-6300 Ilmenau/Thür.



ATARI - Fachhändler
MS - DOS
Kopierer, Drucker, Telefaxgeräte
usw., usw., usw.......

Porzellanstraße 30 · 6300 Ilmenau/Thür. · Tel. 2965

O-7500 Cottbus

COMPUTER & ART
Hard- und Softwarevertriebs GmbH
O-7500 Cottbus
Thälmannplatz 48
Telefon: Cottbus 23696
DTP-Center
ATARI Systemhändler

O-8027 Dresden

Dresdner Computer Center
Ihr erster ATARI System-Fachhändler in Dresden
Kaitzerstraße 82
8027 Dresden Tel.: 47 88 65

ÖSTERREICH

A-1030 Wien

Ihr ST-Fachhändler in Wien
Computer-Studio
Wehsner Gesellschaft m b H
A-1030 Wien
Landstraßer Hauptstraße 2
Hilton-Einkaufspassage

A-1040 Wien

Ihr ST-Fachhändler in Wien
Computer-Studio
Wehsner Gesellschaft m b H
A-1040 Wien · Paniglgasse 18-20
Tel. (02 22) 5 05 78 08, 5 05 88 93

A-1050 WIEN

KNEISZ GmbH
Schönbrunnerstr. 110
1050 Wien
Tel: 55-75-13
Tel: 55-13-80
Fax: 55-29-59/24



ATARI Friends

A-1180 Wien

Ihr kompetentes Atari-Fachgeschäft in Wien
Computing
Tel. (0222) 408 52 56
Fax: (0222) 408 99 78
A-1180 Wien - Schulgasse 63

A-2340 MÖDLING

BESTENS BETREUT
bei
BÖHM Ges.m.b.H.
Ihr
ATARI Systemfachhändler
! 2 Jahre Garantie !
A-2340 Mödling, Hauptstr. 10
☎ 02236-86230 - Fax: 25035
DER WEG ZU UNS LOHNT SICH

A-2700 Wr. Neustadt

BESTENS BETREUT
bei
BÖHM Ges.m.b.H.
Ihr
ATARI Systemfachhändler
! 2 Jahre Garantie !
A-2700 Wr.Neustadt, Bahng.42
☎ 02622-20151
DER WEG ZU UNS LOHNT SICH

SCHWEIZ



Ihr Computer Spezialist

5000 Aarau, Bahnhofstrasse 86,
Tel. 064/22 78 40
4102 Basel-Binningen, Kronenplatz,
Tel. 061/47 88 64
5430 Wettingen, Zentralstrasse 93,
Tel. 056/27 16 60
8400 Winterthur, St. Gallerstrasse 41,
Tel. 052/27 96 96
8021 Zürich, Langstrasse 31,
Tel. 01/241 73 73
Grösste Auswahl an Peripherie, Software, Literatur und Zubehör.

CH-1700 Fribourg

FRIDAT SA INFORMATIQUE
ehem. Softy Hard's Computershop

VOTRE SPECIALISTE
Rte des Grives 4
1700 Granges-Paccot/Fribourg
Tel. 0041 (0)37 26 66 28
Fax. 0041 (0)37 26 61 06

CH-2503 Biel

URWA ELECTRONIC
Computer Hard- und Software

Ihr ATARI ST Spezialist
in der Schweiz.
☎ 032/41 35 35

Bözingenstraße 133, 2504 Biel

CH-3072 Ostermundingen



ATARI
NEC · CITIZEN · EPSON · HEWLETT PACKARD · Computer Drucker

cad computer atelier dellsperger
Zentrum – Alpenstrasse 1
CH–3072 Ostermundigen
Telefon: 031 31 00 32

CH-4313 Möhlin

BCR Computerdienst
Bahnhofstrasse 63
CH-4313 Möhlin

Computersysteme
EDV-Beratung
Installationen
CAD Anlagen
Datenpflege + Service

ATARI · NEC · star

Tel. 061 88 30 32 FAX 061 88 30 03

CH-4500 Solothurn

Fluri Informatik
Hard- & Software, Zubehör
ATARI Schulungszentrum
Desktop Publishing
Systemlösungen

Schänzlistr. 4 4500 SOLOTHURN 1
Tel. 065 / 23 68 58 Fax. 065 / 23 16 57

CH- 4625 Oberbuchsiten

STECTRONIC M. Steck
Electronic-Computer-Shop

Hauptstr. 104/137
CH-4625 OBERBUCHSITEN
Tel. 062/63 17 27 + 63 10 27

CH-4665 OFTRINGEN

Warum nicht gleich zum Fachmann?

IHR ATARI PARTNER

Unser Computerzubehör

Drucker	von	Panasonic, Epson
Plotter	von	HP, Seikosha
Monitore	von	Epson, Matrix, Sony
Scanner	von	Panasonic, Epson,
Festplatten	von	Atari, Vortex,
Belichter	von	Ultre

Unsere Dienstleistungen

Beratung		Verkauf
Installation		Schulung
Service		Reparatur
Scann	–	Druckservice
Plott	–	Belichtungsservice

PYRAMID
PYRAMID electronics AG
Industrie Kleinfeld 33
4665 Oftringen
Tel: 062/97 33 55 Fax: 062/97 30 79

LUXEMBURG



Inserieren in der ST-COMPUTER bringt Erfolg.
Über 100.000 Druckauflage

Heim Verlag

Kleinanzeigen

Biete Software

Anwender- u. Spielesoftware
Laufend n. Programme a. Lager
Riesenauswahl an Original
Soft- u. Hardware
Gratiskatalog anf. o. anrufen
W. Wünsch
Soft- u. Hardware,
(07231) 766595
Friedenstr. 212, 7530 Pforzheim

---

* Preissenkung!!! *
* Softstation *
* 2 fortl. PD von ST-Comp. (1/2, usw) *
* oder 1 PD ab Nr. 225 DM 4, *
* bel. komp. und sonst. Serien DM 5, *
* Disk. wahlw. blau, weiß, rot, grün, *
* gelb od. orange *
* Porto + NN DM 6, *
* Tel.: (07195) 53707 *

Dateiprogramm speziell für Schallplatten und Cassetten, schnell und bedienungsfreundlich. Vollversion 50,–, Demo für 200 Platten 10,– oder Info kostenlos. Ludwig, Dorfstr. 1, 8031 Alling-Biburg

---

Leonardo-ST Vers. 1.0 DM 60
S.T.A.D. Vers. 1.3+ DM 100
That's Pixel/Harlekin je DM 80
Tel: (04921) 51188 nach 18 Uhr

MODELLBAHNSTEUERUNG MIT DEM ST mit Fahrplänen oder direkt im Gleisbild, Fahrstr. Anfahrverz., Ansteuerung im Interrupt...
Info bei: Martin Birn,
Knoblochstr. 51, 7100 Heilbronn

PD-Soft 2 DM inkl. Disk. **Katd. 0 DM** eig. 1,20 DM. J.v.Amstel, Petersstr. 37, 4179 Weeze, BRD

BAUSTATIK FEM-Platten, EB. Stabwerksprg, Dipl.-Ing. U. Precht Hüttenkamp 11, 4970 Bad Oeynhausen 1,
Tel.: (05222) 82018

HP-9000 3,5"-Diskette (LIF) im ST-Laufwerk lesen + beschreiben (ASCII-Datenaustausch, HP-86 lesen); auch für HP-150; Info: Dr. M. Poech
Tel. (0711) 297149, 18–21 Uhr

**Public-Domain-Software
ST-Computer, GFA, XEST usw.
je 3,40 DM inkl. 2S/2D-Disk
Lieferung innerh. 24 Stunden!
Verzeichnisse sind lieferbar
Tel.: (04134) 8689**

STATT vieler Worte! Leo's-PD-CORNER! Für nur **1,– DM** erhalten Sie die **neueste Katalog-Disk!**
Goldberg 38, 3559 Allendorf

SAMMLUNG VON PD-DISKETTEN DER ST-COMPUTER AB NR. 1 BIS 333 FÜR 630 DM ZU VERKAUFEN
J. TWARDOCH WENKENSTR. 67, 4902 BAD SALZUFLEN

WORDPERFEKT 220,- (06431) 41122

DAS ASTROLOGIE-PROGRAMM! Exakte Zeichnung, umfangreiche Wahlmöglichkeiten, Interpretationen für gewerbl. Einsatz!
Tel. (06151) 23720

Orig. Software für Atari ST:
Signum!2 (250,–), STAD (100,–) od. Signum + STAD (300,–), GFA-Draft (150,–), ST-Math (100,–) Salix Prolog (150,–), Simula-Compiler (100,–)
Tel. (09103) 8492

Cashflow Preis VS
Tel. (09232) 7965

PD-Soft ATARI AMIGA PC
Softwarekatalog gratis!!! Bei:
Brunosoft/Schreiber
1000 Berlin 51, Sommerstr. 37

Langläufer! Trainings-Tagebuch mit umfassenden Auswertungsmöglichkeiten auf dem ST für DM 30,- b.
M.Klessing,
Am Schichtmeister 119, 5810 Witten

PD der S-, P-, V-, J-Reihe ab 1,95 DM, Katalog gg. 1,40 DM in Briefmarken, Hintermeier, Postfach, 3551 Lahntal-Sarnau, (06423) 6413

Alle PDs aus ST-Computer für **DM 2,– je Disk**, 10 Disks: 13,– Versandk. + 4,– (Scheck), +7,– (NN). PD-Point Oelde, A. Wessolek, Schorlemer Straße 55, 4740 Oelde

---

Grafiken u. Bilder für Signum u. STAD inkl. 3 Disketten u. Katalog, zum schnellen Auffinden. DM 28,–
Tel. (040) 7119355

BTX-Manager und BECKERbase ST (Original-Prg mit Handbüchern) günstig abzugeben, Tel. (06371) 15681

Verk. PDs Nr 1-363 kompl. für DM 140,-
Tel. (0631) 29707

Die aktuellen Versionen von Signum2 398,– Stad 155,– Creator 219,– HD-Utility 60,– Flexdisk 60,– Neodesk 85,– LaserC mit Deb. 305,– Megamax Modula 350,– Script 175,–
Tel. (0631) 29707

+++ ST:Public Domain Software! Riesenauswahl, Minipreise. 3,5" 2DD Disk ab 2,40 DM. Katalog gratis. Ollis PD-Versand, Goethestraße 6, 6702 Bad Dürkheim.+++

---

oooooooooooooooooooooooo
o PD-Disks aus ST-Compu.(S) und o
o PD-Journ.(J) je 2,5 DM/Abo 2 DM/ o
o Porto: VK 4 DM od. NN 8 DM/ o
o Katalog-disk 4 DM bei: S. Heigert, o
o Berndesallee 6, 6501 Heidesheim, o
o Tel: (06132) 5461 (18-20 Uhr) o
oooooooooooooooooooooooo

Orig. ATARI TOS 1.4 (2 ROMs) 135,– orig. WordPerfect (unbenutzt!) 375,–;
Tel. (0431) 16433

SPC MODULA-2 Sprachsystem V2.0 für ST, lizenziert, DM 275,
Michael Heym, Tel.: (0721) 856329

Signum2 DM 250,–, Stad 1.3+ DM 100,–, PC-Ditto 3.96 DM 100,–, Twenty Four Pro III DM 200,–, Synthworks D50 DM 150,–, alles kompl. DM 500,–
07031/802181

CNC-SIMULATOR
+simulierter Original-Tastatur
W.Schulz, Schlesierstraße 1
6701 Birkenheide
Modul.2 DM 95, Demo DM 25

EIGENES GESCHÄFT? Wir geben unseren Soft- u. Hardwarehandel aus persönl. Gründen zum 31.12.90 auf. PD-Sammlung (1500 Disk für Atari ST) aus ST-Computer, ATARImagazin, XEST, PD-Journal (Reihen J,D,V), GFA-Club, PD-Pool für 3700,– DM. Zusatzangebot: PD's + 2200 Kd-Adressen mit Faktura + PD-Katalog auf Diskette + Anschreiben aller Kd. durch uns, 4900,– DM. CHIFFRE

Lohnsteuer Einkommenst.f. 1990 selbst berechnen. DM 40 Info: Schwabensoftware Kugelplatz 32 8850 Donauwörth Tel. (0906) 5503

Biete Hardware

TV-Modulator ST ab 99,– DM
Info / Bestell.: (02366) 41014

Speichererweiterung für ATARI ST
2.5 MB: DM 450,– 4 MB: DM 880, Einbau + 50,– DM, mit Garantie
heiler engineering (0721) 376622

Megafile 60 (9Mon) 850,–, Farbmonitor SC1224 (wie neu) 350,–, Technobox Drafter (neu) 500,–, Calamus 1.09N 400,–, Becker CAD 99,–
Volker Misselhorn, Lenaustraße 1,
3300 Braunschweig, T. (0531) 55607

MMU, GLUE, DMA je 90 DM, SHIFTER, BLITTER je 110 DM, TOS 1.0/1.2 30 DM, auch andere Teile von 520 ST; 16MHz, pro VME, 8K Cache 380 DM, Maus 40 DM, Tel. (069) 5071694

Verkaufe eine 44MB Wechselplatte von SyQuest für 900,– DM. Verkaufe eine 8MB Quantum Festplatte für 900,– DM. 64x1MB RAMs, das Stück für 9,50 DM Tel. (030) 6184457 ab 19 Uhr.

MEGA ST2, SM124, Maus, Tastschutz + Softw. VB 1600,– T. (040) 4808143

ATARI TT 030/4 10% unter Neupreis inkl. MWSt mit Monitor PTC 1426; 4 MB, 48,6 MB Festplatte 6748,– DM
Tel. (09433) 6539

MEGA ST4, SM124, Maus, Softw.
520 ST, SM 124, Maus, Laufwerk
von 17-19.00 Uhr, (08721) 5343

ATARI MEGA ST4, SM124, VB 2500,– DM; Tel. (02565) 2718

TOS 1.4 mit Patches, 120 ns EPROMs - 59 DM T. (0212) 208613
EPROM-, GAL-Programmiergerät je 129 DM, 2MB Platine 35 DM

TOS-komp. Festpl., 44MB form. 28 ms, OMTI-Contr., OMTI-Adapt. orig. Treiber-Softw., c–t-Artikel, Netzteil, Kabel, zum Einbau DM 400,–
Tel. (0241) 31165

PC-Speed V1.4 DM 298,–; **Maus**, kaum benutzt, DM 70,–; **TOS 1.4** in 6 ROMs, gepatcht, DM 50,
Tel. (089) 605493 ab 18.00 Uhr

TOS 1.4 mit Patches nach ST 1/90
DM 80,–, Tel. (0202) 590754

RainbowTOS 1.4 mit ST-Computer Patches 70,– **Schaltpläne ST/ MEGA/SM124** je 25,– **3M Byte** Erweiterung 450,– (04351) 44252

ATARI ST + Casio FX-850P Rechnerkopplung 90 DM / **Systembuch** FX-850P (Schnittstellenschaltpl. u.a.) 22 DM / Info v. Schleitzer & Krumm,
Voßstr. 45, 3 Hannover

TOS 1.4 gepatched nach ST-Computer 1/90 + 10/90 in 6 ROMs für 60 DM bei Vorkasse, sonst +Nachnahme: Postfach 3345, 2900 Oldenburg

---

MEGA-ST-Tastaturen, Floppies, Netzteile, Mäuse neu u. original.
Rauch (07031) 52607

Einige **MEGA-ST-Leergehäuse** zum Einbau von Festplatten usw.
Rauch (07031) 52607

---

2 Hypercache (350,– u. 500,–). Einige **SM 124**, neu (280,–), gebr. (150,–)
Rauch (07031) 52607

1040ST, SM124, PC-Speed, SF314, Laufw. 1,44MB, Festp. 30MB, VB 1950 DM Farbmon., 240 DM, Akkustikkoppler (BTX), 190 DM, Tel: (06151) 144150, 18-22 Uhr

---

NEC P7+ 1 J. alt mit dt. Handbuch 999 DM, Michael (07144) 38203

520STM, 4MB, TOS 1.4, MEGA-Clock, PC-Speed 1.4, Spectre GCR, SM124, 2xSF314, 1xSF354, Harddisk Vortex HD-Plus 60MB, Maus, Literatur
Nur komplett, 3500,– D. Adam
Tel. (089) 857839 68 od. (089) 7005137

MEGA ST2, SM124, **Harddisk** HDplus 20, **AT-Speed**, Grafikerw. Overscan + div. Software/Bücher für **2000,– DM** Tel. (08363) 8891

Lacom Harddisk 85 MB, neuw., SCSI, 950 DM, Tel. (0561) 403918

Videodigitizer „Easytizer" für 200,– Tel. (0931) 409679

ATARI ST Ram-Erweiterung um 2MByte, Platine (industriell hergestellt), Zubehör, Einbauanleitung (ohne Rams) für 79,– DM (V-Scheck) inkl. Porto und Verpakkung. J. Becker, Jülicherstraße 46-48, 5100 Aachen Tel. (0241) 505953 (24h)

DIN-A-4-Handy-Scanner für ST 400 dpi, VB 1100 DM mit Bearbeitungssoftw. Tel. (0941) 997860

Floppy NEC FD1037 - 129,– DM, Geh. dazu - 19,– DM, Sig.-Kabel - 16,– DM, anschlußfertig - 189,– DM,
Tel. (089) 806823

Suche Software

Imagic (nur Original) von ASH
Tel. (05069) 8297

Suche Hardware

Suche HANDY SCANNER gebraucht mit Software Tel. (09324) 3341

Suche Schrott ST's zwecks Ausschlachtung. Schreibt mit Preisvorstellung und Fehlerbeschr. an Wolfgang Wüsten, Winkelsfeld 10, 4056 Schwalmtal/Amern

Kontakte

Suche TeX User, der Erfahrung mit TeX auf dem Atari hat. Selbst bin ich Anfänger mit TeX. Tel.: (02821) 27050

Verschiedenes

**Calamus-Laserdruck ab 40Pf/S
Scan-Service DIN-A-4 8DM/S
Desktop-Publ. Tel. 0941/90556**

An Programmierer u. Softwarehäuser! Sachbuchautor (Profi) schreibt oder korrigiert Ihre Handbücher. R.D. Zens (02266) 44228

+++++++++++++++++++++++++++++
+ RAM-Erweiterungen für Alle! +
+ SPEICHER - SPEICHER +
+++++++++++++++++++++++++++++
+ Das ganze SIP-Speicherpro- +
+ gramm von Sven Betz gibt's +
+ auch bei mir. Sogar noch +
+ billiger: +
+++++++++++++++++++++++++++++
+ (Infos über die winzigen +
+ SIP-Erweiterungen finden +
+ sich in der ST-Computer +
+ 11/90, S.97 u. 09/90 im +
+ Kleinanzeigenteil) +
+++++++++++++++++++++++++++++
+ 2,5 MB für alle ST. 449,– +
+ (auch 260er u. ST/FM) +
+ 4 MB für alle ST 738,– +
+ MEGA 1 auf 2MB 339,– +
+ 1040er (neuere) 2MB 339,– +
+ ST/E SIMMs auf 2MB 249,– +
+ ST/E SIMMs auf 4MB 478,– +
+++++++++++++++++++++++++++++
+ 260er, 520er auf +
+ 1 MB (steckbar auf +
+ 2,5 oder 4MB !!) 188,– +
+++++++++++++++++++++++++++++
+ Die anderen Hardware-Teile +
+ (Turbo 16, AT-Once usw.) +
+ baue ich für 10 DM günstiger +
+ ein als in den aktuellen +
+ Anzeigen von Sven Betz +
+ inseriert (s. diese Ausgabe) +
+++++++++++++++++++++++++++++
+ z.B. AT-Once inkl. +
+ Einbau für 493,– +
+ Turbo 16 + AT-Once f. 1049,– +
+++++++++++++++++++++++++++++
+ Joachim Lühr, Adlerstraße 20 +
+ 2000 Hamburg 60 +
+ Tel. (040) 6907469, +
+ am besten Mi., Fr. o. Sa +
+++++++++++++++++++++++++++++


Programmier Praxis
INHALT
Jump - Marken für Turbo C
C und Assembler .... Seite 72
Gedrehte Bildschirmausschnitte
GFA-BASIC .... Seite 83
Der etwas andere Zugriff
Assembler .... Seite 87
Resource-Einbindung in C
C .... Seite 90
Farbpalettenwechsel
MAXON-Pascal .... Seite 97

JUMP

DEM TURBO C-EDITOR AUF DIE SPRÜNGE GEHOLFEN

Stefan Dreckmann

WIR SCHREIBEN DAS JAHR 1990. DIE ZEITEN UNBEQUEMEN PROGRAMMIERENS MIT LAHMEN EDITOREN OHNE JEDEN KOMFORT SIND ENDGÜLTIG VORBEI. IN GANZ DEUTSCHLAND HAT SICH DIE ERKENNTNIS DURCHGESETZT, DASS EINE MARKENFUNKTION SO SELBSTVERSTÄNDLICH ZUR AUSRÜSTUNG EINES QUELLTEXTEDITORS GEHÖRT WIE EINE BLOCK- ODER SUCHFUNKTION. IN GANZ DEUTSCHLAND? NEIN! TIEF IM SÜDEN DER REPUBLIK WEHRT SICH EIN KLEINES HÄUFLEIN UNENTWEGTER STANDHAFT GEGEN DIE ZEICHEN DER ZEIT UND DIE ANSPRÜCHE DER BENUTZER.

Ihr Produkt, Turbo C, ist in jeder Beziehung erstklassig: schneller, kompakter Code, kurze Turnaround-Zeiten, optimale Zusammenarbeit zwischen Editor, Compiler und Linker. Erstklassig in wirklich jeder Beziehung? Nein! Der Editor ist lahm wie weiland Wordplus. Und - er besitzt keine Markenfunktion!

Diesem bedauernswerten Zustand wird jetzt (endlich!) ein Ende gesetzt, denn Jump ist da. Jump hilft TC auf die Sprünge. Mit Jump kann man neun Marken beliebig im Quelltext definieren und blitzschnell anspringen. Und Jump springt ebenso schnell an den Ausgangspunkt zurück. Zusätzlich zu dieser konventionellen Markenfunktion kann Jump den Editor veranlassen, die aktuelle Cursor-Zeile in den Zwischenspeicher zu übernehmen. Am Zielort wird diese Zeile dann mit Insert z.B. an eine Liste von Funktionsdeklarationen angefügt. Natürlich ist Jump ein Accessory.

Da die Funktionstasten des ST vom Editor ohnehin ignoriert werden, bietet es sich an, diese für die Marken zu nutzen. Drückt man eine Funktionstaste zusammen mit Shift, merkt sich Jump die aktuelle Cursor-Zeile. Ohne Shift wird eine solche Marke angesprungen. Hält man die Control-Taste gedrückt, wird die aktuelle Zeile in den Zwischenspeicher kopiert und dann gesprungen. Die Taste F10 kann nicht mit einer Marke belegt werden. Hier speichert Jump die Ausgangszeile, damit man bequem dorthin zurückkehren kann.

Alle Funktionen des Accessories basieren letztendlich auf den Funktionen des Editors. Eine zentrale Rolle spielt dabei die *Find line*-Funktion. Sie wird per Control-L aufgerufen, und es erscheint eine Dialogbox, in die die aktuelle Zeile vom Editor eingetragen ist. Jump liest diese Zeilennummer aus und trägt die neue ein, indem es einfach Tastendrücke auf die entsprechenden Zifferntasten simuliert. Schließlich kommt noch die Simulation von Return - und ab geht die Post. Um diese Aktionen auszuführen, muß Jump:

1. einen Druck auf eine Funktionstaste abfangen können,
2. Tastendrücke simulieren können und
3. die Adresse wissen, unter der TC die aktuelle Zeilennummer einträgt, damit diese ausgelesen werden kann.

Tastendrücke

Ein Tastendruck wird üblicherweise mit Hilfe der AES-Funktion *evnt_keybd()* abgewartet. Leider liefert aber das AES Tastendrücke grundsätzlich nur an die Hauptapplikation (also TC in diesem Falle) und nicht an Accessories. Dieser einfache Weg, die Funktionstasten zu überwachen, ist also für Jump verbaut.

Glücklicherweise werden Tastendrücke intern in einem Tastaturpuffer zwischengespeichert. Der Tastaturpuffer wird als Ringpuffer verwaltet, und zwar mit Hilfe zweier „Zeiger“, die jeweils auf die aktuelle Schreib- und Leseposition zeigen. Mit Hilfe der Betriebssystemfunktion *Iorec()* (XBIOS 14) bekommt man einen Zeiger auf die in Bild 1 beschriebene Struktur.

Es gibt drei solcher Puffer im ST, für MIDI, die Tastatur und die serielle Schnittstelle. Die letzten beiden Einträge der Struktur sind nur für die serielle Schnittstelle interessant. Der Aufruf von *Iorec(1)* liefert einen Zeiger auf die Iorec-Struktur für den Tastaturpuffer.

In seiner Hauptschleife überwacht Jump den Puffer, indem es sich einfach *iorec->ibufhd* merkt, ein bißchen wartet und kontrolliert, ob der Wert sich verändert hat. Wenn ja, wurde etwas in den Puffer geschrieben, also eine Taste gedrückt. Jump kontrolliert, ob es sich um eine Funktionstaste handelt, und tritt bei Bedarf in Aktion.

Der Tastaturcode ist immer ein Langwort, und zwar steht im oberen Wort der Scan- und im unteren der ASCII-Code der gedrückten Taste. Bei Funktionstasten ist der ASCII-Code natürlich 0. Sie sind übrigens die einzigen Tasten, bei denen sich der Scancode mit Shift von dem ohne Shift unterscheidet. Jump lauert in *main()* auf Tastendrücke. Da natürlich

auch auf eine Anwahl des Accessories im Menü reagiert werden muß, geschieht dies mit *evnt_multi()*.

Hat man einmal einen Zeiger auf die Iorec-Struktur, ist es ein Leichtes, einen Tastendruck zu simulieren [s. Funktion *tastendruck()* in Listing 1]. Man schreibt einfach den betreffenden Code als Langwort an die Adresse *iorec->ibuf* (Anfang des Tastaturpuffers) und paßt *iorec->ibufhd* und *iorec->ibuftl* entsprechend an, um dem Rest der Welt vorzugaukeln, daß soeben ein Tastendruck stattgefunden hat. Allerdings muß man dann dem Rest der Welt auch Gelgenheit geben, den Tastendruck zur Kenntnis zu nehmen. Solange nämlich ein Accessory seiner Arbeit nachgeht, liegt das Hauptprogramm auf Eis. Erst ein beliebiger evnt...-Aufruf sorgt dafür, daß andere auch mal was erledigen können. In der Funktion *tastendruck()* werden also erst einmal Pausen eingelegt, und zwar so lange, bis der „Tastendruck" auch wirklich angekommen ist. Hier eine konstante Zeitspanne zu warten, ist nicht angebracht. TC braucht nämlich unterschiedlich lange, um auf einen Tastendruck zu reagieren, je nachdem, was es gerade tut. Auf das Problem der Synchronisierung von Jump und TC werde ich an anderer Stelle noch einmal zurückkommen.

```
typedef struct
    {
    void *ibuf;         /* Zeiger auf den Pufferanfang */
    int ibufsiz;        /* Größe des Puffers */
    int ibufhd;         /* aktuelle Schreibposition */
    int ibuftl;         /* aktuelle Leseposition */
    int ibuflow;        /* untere "Wassermarke" */
    int ibufhi;         /* obere "Wassermarke" */
    } IOREC;
```

Bild 1: IOREC beschreibt einen internen Puffer

```
typedef struct
    {
    int ob_next;            /* nächstes Objekt */
    int ob_head;            /* erstes "Kind-"objekt */
    int ob_tail;            /* letztes "Kind-"objekt */
    unsigned ob_type;       /* Art des Objekts */
    unsigned ob_flags;      /* Manipulationsflags */
    unsigned ob_state;      /* Zustand des Objekts */
    OBSPEC *ob_spec;        /* zeiger auf weitere Struktur */
    int ob_x;               /* x-Pos. linke, obere Ecke (relativ) */
    int ob_y;               /* y-Pos. */
    int ob_width;           /* Breite des Objekts */
    int ob_height;          /* Höhe des Objekts */
    } OBJECT;
```

Bild 2: OBJEKT beschreibt ein Grafikobjekt

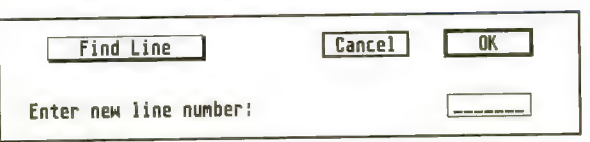


Bild 3: Die "Find Line" Dialogbox des TC Editors

Suche Nummer

Nachdem nun die minderen Probleme erledigt sind, wenden wir uns der Hauptsache zu: Wie erfährt Jump die jeweils aktuelle Zeilennummer? Der Hauptangriffspunkt ist hier, wie gesagt, die Funktion *Find line* des Editors. Ein Aufruf der Funktion ist leicht: Man simuliert einfach einen Tastendruck auf Control-L. Um allerdings zu verstehen, wie die Zeilennummer ausgelesen wird, muß man wissen, was im einzelnen abläuft, wenn man die Funktion aufruft.

Üblicherweise bedient man sich der AES-Funktion *objc_draw()*, um eine Dialogbox, wie sie auch *Find line* benutzt, auf den Bildschirm zu bringen. Dazu übergibt man (unter anderem) die Adresse eines sogenannten Objektbaums, in dem die einzelnen Elemente der Box codiert sind. Das AES zeichnet dann die Box mit Hilfe der im Objektbaum vorhandenen Informationen. Die Standardelemente der grafischen Benutzerführung des ST (also Menüleisten, Dialogboxen, aber auch Fenster) werden intern aus sogenannten Objekten aufgebaut, die einheitlich durch die Struktur in Bild 2 beschrieben wird.

Die meisten Einträge sind in diesem Zusammenhang uninteressant. Wer sich näher über Objekte und Objektbäume informieren will, sei z.B. auf [1] und [2] verwiesen. Ich werde an dieser Stelle nur auf drei Einträge näher eingehen: *ob_type*, *ob_flags* und *ob_spec*.

Es gibt insgesamt 13 Objekttypen. Sie werden durch eine entsprechende Nummer in *ob_type* unterschieden. So ist z.B. der äußerste Kasten des *Find line*-Formulars ein Objekt vom Typ *G_BOX* (Nr. 20), der „Titel" des Formulars und die Eingabezeile gehören zum Typ *G_BOXTEXT* (Nr. 22) usw. Die Eigenschaften der einzelnen Objekte (z.B. SELECTABLE, EDITABLE usw.) sind in *ob_flags* bitweise codiert. So sind z.B. die beiden Buttons, „OK" und „Cancel" anwählbar und Exit-Objekte; „OK" ist zusätzlich noch ein Default-Objekt, da es auch mit der Return-Taste angewählt werden kann. Also sind in den Objektstrukturen für die Buttons die Bits 0 und 2, beim „OK"-Button zusätzlich das Bit 1 gesetzt. Die Eingabezeile ist edierbar, d.h. daß das Bit 3 in *ob_flags* gesetzt ist (siehe Bild 3).

Offensichtlich werden alle Elemente der Dialogbox von der äußeren Box umfaßt; sie alle sind „Kind"-Objekte dieser äußeren Box. Die Adresse der äußeren Box ist die Anfangsadresse des gesamten Objektbaums. Dort fängt das AES beim Zeichnen an und „hangelt" sich per *ob_next* bzw. *ob_head* durch den gesamten Baum, um alle Objekte zu zeichnen.

Hat man erst einmal die Anfangsadresse des Baums ermittelt, kann man ebenso verfahren, bis man ein Objekt gefunden hat, das EDITABLE ist. Da es in diesem speziellen Fall nur ein solches Objekt gibt, ist damit das Objekt gefunden, in dem sich die Eingabezeile befindet. Allerdings hat man damit immer noch nicht den Zeilen-String selbst. Dessen Adresse ist nämlich nicht direkt in die Objektstruktur eingetragen, sondern muß indirekt über *ob_spec* ermittelt werden. Die Einträge in *ob_spec* sind je nach Objekttyp verschieden. Bei dem hier vorliegenden Typ *G_BOXTEXT* zeigt *ob_spec* auf eine weitere Struktur (Bild 4).

Der erste Eintrag der TEDINFO-Struktur ist endlich die Adresse des gesuchten Eingabe-Strings.

Die Eingabezeile wird im zweiten Teil der Funktion *get_inputAdr()* ermittelt. Wie Sie sehen, wird die While-Schleife abgebrochen, sobald ein edierbares Objekt gefunden ist. Die Anfangsadresse des Baums wird aus der globalen Variablen *TCobjAdr* (Zeiger auf OBJECT) übernommen. Um zu verstehen, wie diese ermittelt werden kann, muß man noch ein wenig tiefer in die Eingeweide des Betriebssystems vordringen.

Ins wilde GEMDOS

Bekanntlich ist das Betriebssystem des ST in zwei Teile gegliedert, das TOS und das GEM. TOS interessiert hier nicht weiter, wohl aber GEM, das seinerseits in zwei Teile zerfällt, das VDI und das AES. Das AES stellt dem Programmierer unter anderem eine umfangreiche Sammlung von Funktionen zur Darstellung und Verwaltung grafischer Objekte zur Verfügung. Da TC leider nicht freiwillig die Adresse der *Find Line*-Dialogbox herausrückt, muß man sie sich selbst direkt vom AES besorgen. Der Trick dabei ist, daß TC die Adresse ja ans AES übergeben muß, um die Dialogbox zeichnen zu lassen. Hier schmuggelt sich Jump hinein und liest sie aus, bevor das AES mit seiner Arbeit beginnt.

Beim Aufruf einer AES-Funktion müssen dieser jede Menge Parameter übergeben werden. Dies geschieht nicht über den Stack (wie bei allen Funktionen des TOS), sondern mit Hilfe von sechs Feldern, die vor jedem Aufruf entsprechend zu belegen sind. Die Adressen der Felder werden in den sogenannten AES-Parameterblock eingetragen. In Turbo C gibt es keinen *AESPB*, vielmehr wird für VDI und AES zusammen intern der sogenannte *GEMPARMBLK* verwendet (Bild 5).

```
typedef struct
    {
    char *te_ptext;      /* Zeiger auf den Text */
    char *te_ptmplt;     /* Zeiger auf die Textmaske */
    char *te_pvalid;     /* Zeiger auf die Typmaske */
    int te_font;         /* Zeichensatz */
    int te_resvd1;
    int te_just;         /* Justierung des Textes */
    int te_color;        /* Farbe des Rechtecks */
    int te_resvd2;
    int te_thickness;    /* Rahmendicke */
    int te_txtlen;       /* Länge des Textes */
    int te_tmplen;       /* Länge der Textmaske */
    } TEDINFO;
```

Bild 4: Die "Textinformationsstruktur"

```
typedef struct
    {
    long cb_pcontrol;    /* Zeiger auf control[] */
    long cb_pglobal;     /* Zeiger auf global[] */
    long cb_pintin;      /* Zeiger auf int_in[] */
    long cb_pintout;     /* Zeiger auf int_out[] */
    long cb_padrin;      /* Zeiger auf adr_in[] */
    long cb_padrout;     /* Zeiger auf adr_out[] */
    } AESPB;
```

Bild 5: Der AES-Parameterblock

Alle Felder müssen vom aufrufenden Programm zur Verfügung gestellt werden. Programmiert man in einer Hochsprache, bemerkt man davon allerdings meistens nichts, da der Compiler das erledigt. Das *global[ ]*-Feld wird vom AES [bei Aufruf der Funktion *appl_init()*] belegt. Die Felder *Control[ ]*, *int_in[ ]* und *adr_in[ ]* dienen zur Übergabe von Werten an das AES, während Rückgabewerte von diesem in *int_out[ ]* bzw. *adr_out[ ]* geschrieben werden. Dabei sind die int...-Felder für Integerwerte und die adr...-Felder entsprechend für Adressen vorgesehen. Das Feld *Control[ ]* enthält unter anderem den Opcode für die gewünschte AES-Funktion.

Nachdem ein Programm die Felder belegt und deren Adressen in den AES-Parameterblock eingetragen hat, wird dessen Adresse in das Register D1 geladen. In D0 muß ein Opcode eingetragen werden, und zwar 115 beim Aufruf von VDI-Funktionen und 200 für das AES. Jetzt erst erfolgt der eigentliche Aufruf des AES, und zwar (wie bei allen Betriebssystemfunktionen) über einen sogenannten „Trap" (engl. „Falle"), hier Trap 2.

Fallen im ST

Traps sind waschechte Maschinenbefehle. Wenn die CPU in eine solche Falle rennt, schaltet sie zunächst in den Supervisormodus und rettet PC und Statusregister auf den Stack. Dann wird der PC mit einer neuen Adresse geladen. Diese neue Adresse ist in einem Block von Systemvariablen zu finden, der sich ganz am Anfang des RAM befindet. Normalerweise zeigt die betreffende Variable auf den Trap Dispatcher im ROM, also auf den Teil des Betriebssystems, der die GEM-Systemroutinen verwaltet. Da aber die Adresse selbst im RAM liegt, kann man sie ändern (den Exception-Vektor „verbiegen") und auf eine eigene Routine zeigen lassen.

Genau dies macht Jump in der Funktion *call_formular()*. Dazu bedient es sich der Routine *Setexc()* (BIOS 5). Ihr werden die Nummer (nicht die Adresse!) des zu ändernden Exception-Vektors und der neue Vektor übergeben, der dort eingetragen werden soll. Man erhält den alten Vektor zurück, der bei Jump in der globalen Variablen *trapZwoAdr* gesichert wird. (Alle Systemvariablen und die Nummern der Exception-Vektoren sind übrigens z.B. in [1] zu finden)

In *call_formular()* ändert Jump also zunächst den Exception-Vektor für den Trap 2, der für AES- und VDI-Aufrufe zuständig ist. Dann wird ein Tastendruck Control-L simuliert. Der Editor schreibt daraufhin zunächst die aktuelle Zeilennummer in die *Find Line*-Dialogbox und versucht dann, sie auf den Bildschirm zu bringen, nach dem oben beschriebenen Muster. Allerdings wird jetzt zunächst nicht der Trap Dispatcher, sondern die kleine Maschinenroutine angesprungen, auf die Jump den Vektor verbogen hat. Diese kleine, private Routine [*kuckma_rein()*] sieht sich zunächst den Inhalt des Registers D0 an und stellt anhand des Opcodes fest, ob eine AES-Funktion aufgerufen werden soll. Falls ja, wird die Adresse des AESPB von D1 nach A0 geladen. Da das erste Element der Struktur die Adresse des *Control[ ]*-Feldes ist, gelangt mit

```
move.l (a0),a1
```

dessen Adresse nach A1. In Control[0] muß der Opcode der aufzurufenden Funktion eingetragen sein. Jump interessieren nur zwei Funktionen: *objc_draw()*(Opcode 42) und *form_do()*(Opcode 50).

Beim Aufruf dieser Funktionen muß sich in *adr_in[0]* ein Zeiger auf die Adresse des betreffenden Baums befinden. Da A0 die Adresse des AESPB enthält und *adr_in[ ]* der vierte Eintrag in der Struktur ist, kopiert

```
move.l 16(a0),TCobjAdr
```

den Zeiger in die globale Variable *TCobjAdr*. Danach wird das Flag *objcDraw* gesetzt. Falls die Routine *form_do()* aufgerufen werden sollte, wird nur das Flag *formDo* gesetzt. Wozu die beiden Flags gebraucht werden, erkläre ich gleich. Nachdem *kuckma_rein()* seine Arbeit getan hat, verzweigt es zur Originaladresse des Trap Dispatchers, die bekanntlich in *TrapZwoAdr* zu erfahren ist. Dort wird dann fortgefahren, als sei nichts passiert.

Synchronisierung

Warum wird aber in *call_formular()* nach dem Tastendruck in einer Schleife auf die Reaktion des Editors gewartet? Genügt es nicht, einfach einmal *evnt_timer()* aufzurufen? Das Problem der Synchronisierung der beiden Programme habe ich bei der Funktion *tastendruck()* schon kurz erwähnt. Der Editor braucht unterschiedlich lange, um die Dialogbox auf den Bildschirm zu bringen. Bei sehr langen Quelltexten kann es schon eine ganze Weile dauern, bevor sie endlich erscheint. Natürlich

wäre es möglich, einfach sehr lange zu warten, nachdem der Tastendruck simuliert wurde. Aber wie lange wäre das? Legte man eine bestimmte Zeit fest, würde in den meisten Fällen vermutlich zu lange gewartet, aber in einigen Fällen zu kurz. Die Lösung mit dem Flag *objcDraw* synchronisiert Jump und den Editor optimal, solange man mit einem guten Wert für Pausen sicherstellt, daß nicht zu viele *evnt_timer()*-Aufrufe stattfinden, denn die kosten ja auch Zeit. Außerdem: was würde passieren, wenn man Jump einschaltete, ohne sich tatsächlich in der Entwicklungsumgebung zu befinden? Die Adresse *TCobjAdr* wäre ungültig, der Versuch, über sie die Adresse der Eingabezeile zu erfahren, würde mit einiger Sicherheit zum Absturz des Systems führen. Das hier angewandte Verfahren stellt einigermaßen sicher, daß so etwas nicht passieren kann. Denn nur, wenn eine Dialogbox gezeichnet wurde, wird die Funktion bis zum Ende fortgesetzt. Wird aber *MAX_GEDULD* überschritten, gibt es Proteste.

Aus ähnlichen Gründen wird in *call_formular()* ein zweites Mal in einer Schleife gewartet. Es dauert nämlich verschieden lange, bis die Dialogbox gezeichnet ist, abhängig davon, ob ein Blitter vorhanden ist oder nicht. Sobald aber durch *kuckma_rein()* das Flag *formDo* gesetzt ist [*TC hat also form_do()* aufgerufen], kann man sicher sein, daß die Box tatsächlich fertig gezeichnet wurde. Andernfalls würde Return ignoriert werden, und Jump käme ganz schön ins Schwitzen.

Eine Zeile kopieren

Die Kopierfunktion von Jump [*zeile_merken()*] macht sich die Tatsache zunutze, daß sich eine Zeile, die mit Control-Y gelöscht wurde, danach im Zwischenspeicher des Editors befindet. Also muß man die gewünschte Zeile nur löschen, sie mit Insert an derselben Stelle wieder einfügen und schon hat man erreicht, was man wollte. Hier allerdings scheint es wieder ein Synchronisationsproblem zu geben. Ich muß gestehen, daß mir dazu bisher noch keine vernünftige Lösung eingefallen ist. Zwar ruft TC nach dem Befehl Control-Y reichlich VDI-Funktionen auf. Man sollte also meinen, eine Synchronisation, wie sie bei der *Find Line*-Dialogbox so tadellos funktioniert, wäre auch hier brauchbar. Tatsächlich aber klappt der Trick mit einer *kuckma_rein*-Routine aus unerfindlichen Gründen hier nicht. Also bleibt nur, statt der kleinen Pause eine längere Pause einzulegen. Leider funktioniert dies nicht immer; in seltenen Fällen gelangt die betreffende Zeile nicht in den Zwischenspeicher. Es ist mir allerdings noch nie passiert, daß zwar Control-Y, aber nicht Insert funktioniert hätte, so daß also nicht zu befürchten ist, daß plötzlich eine Zeile im Quelltext verschwunden wäre. Man hat lediglich eine Zeile zuviel. Sollte das passieren, genügt es, einfach die Undo-Taste zu benutzen. TC springt dann wieder an den Ausgangspunkt zurück und löscht die überflüssige Zeile. Wem diese (wie ich hoffe einzige!) Schwachstelle des Programms nicht behagt, der kann die Kopierfunktion auch einfach weglassen und die Zeilen „von Hand" kopieren. Ich persönlich benutze die Kopierfunktion ausgiebig. Wenn es einmal nicht klappen sollte, fasse ich mich in Geduld und versuche es einfach noch einmal.

Finger weg von Fremdprogrammen

Schließlich noch ein paar Worte zu der Funktion *autoswitch()*. Da man direkt aus der Entwicklungsumgebung heraus fremde Programme starten bzw. das gerade in Arbeit befindliche testen kann, ist es notwendig, dafür zu sorgen, daß sich Jump in deren Arbeit nicht einmischt. Also muß das Accessory immer dann in den Ruhestand gehen, wenn ein weiteres Programm gestartet wird. Außerdem darf natürlich auch nach dem Verlassen der Entwicklungsumgebung nicht mehr auf Funktionstastendrücke reagiert werden.

Glücklicherweise verschickt das AES beim Start und bei Beendigung eines Programms an jede Applikation eine Meldung (*AC_CLOSE*). Jump muß also nur auf diese Meldung reagieren und seinen Zustand (*accEin*) entsprechend ändern. Wenn man allerdings ein Programm beendet, das per „Execute" gestartet wurde, gibt es nicht ein, sondern zwei *AC_CLOSE*. Außerdem könnte ein leichtsinniger Benutzer auf die Idee kommen, Jump in einem per „Execute" gestarteten Fremdprogramm zu starten, was mit Sicherheit viel Verwirrung (oder vielleicht auch Bombenstimmung) anrichten würde. Dies ist trotz der Sicherheitsabfrage vor dem Einschalten von Jump möglich, da *appl_find()* bei Programmen, die von anderen Programmen aus gestartet wurden, den Namen des „Eltern"-Programms und nicht den des gerade aktuellen findet. [Ähnliches gilt auch für die Beendigung von TC, solange noch kein anderes Programm gestartet wurde: im Desktop findet *appl_find()* TC immer noch - fehlerhafterweise]

Um also zu gewährleisten, daß Jump wirklich nur da wirkt, wo es auch nützlich ist, wartet es nach dem ersten *AC_CLOSE* (falls es eingeschaltet war) in *autoswitch()* auf eine weitere Nachricht. Ist diese wiederum *AC_CLOSE*, wird per *evnt_multi()* noch ein bißchen gewartet (Verlassen eines Programms, das mit „Execute" gestartet wurde), und erst dann wird der normale Geschäftsbetrieb wieder aufgenommen. Wie man sieht, ist es möglich, Jump auch in fremden Programmen einzuschalten, indem man Control-Shift-Alternate gedrückt hält, während man es in der Menüleiste anklickt. Diesen NOTSTART hatte ich ursprünglich nur zum Testen vorgesehen, lasse ihn jetzt aber im Programm, man weiß ja nie. Wer Jump allerdings NOTSTARTet, sollte vorsichtig damit umgehen – ich will hinterher keine Klagen hören!

Jump++

Es ist möglich, Jump an einigen Stellen zu erweitern. So könnte man z.B. verhindern, daß die Dialogbox überhaupt gezeichnet wird, denn die Grafik ist mit Sicherheit der langsamste Teil der Funktion *Find Line*. Dazu müßte man im Assembler-Teil bei Aufruf von *objc_draw()* nach der Übernahme der Adresse und dem Setzen des Flags zunächst *int_out[0]* mit einem Wert ungleich Null belegen und dann die Exception mit „RTE" beenden. [In der Entwicklungsumgebung für den Assembler den Schalter „Privileged Instructions" (-S) setzen!] Dann sollte man allerdings auch *form_do()*-Aufrufe abfangen. Dazu wird der Index des OK-Buttons nach *int_out[0]* geschrieben, die alte Zeilennummer in einen String, eventuell eine neue Zeilennummer in die TEDINFO-Struktur kopiert und wieder „RTE".

Außerdem wäre es sicher nützlich, nicht einen, sondern mehrere Sets von Marken mit Jump zu verwalten (wie wäre es mit sechs; für jedes Fenster, das man in der Entwicklungsumgebung öffnen kann, eins). Auf diese Weise könnte man bei Programmprojekten, die aus mehreren Modulen bestehen, für jeden Quelltext (jedes Fenster) ein eigenes Set von Marken anlegen, das gewechselt wird, wenn man das Fenster wechselt.

Schließlich könnte man noch Funktionen zum Laden und Sichern der Marken implementieren, damit man sie nicht bei jeder Arbeitssitzung wieder neu definieren muß.

Ich habe ein Jump2, das diese Funktionen beherrscht, bereits geschrieben. Allerdings ist das Listing fast fünfmal so lang wie dieses hier. Die Tastenbelegung

ist wesentlich unübersichlicher geworden, denn die Befehle („Set laden", „Set sichern", „Set wechseln" usw.) müssen natürlich auch über Tasten eingegeben werden können. Außerdem gab es Probleme mit dem Redraw, wenn die Fenster nicht Bildschirmgröße hatten, die zu lösen wieder eine Menge an Code erfordert hat.

Insgesamt, denke ich, ist diese einfache Version von Jump schon eine große Erleichterung im Umgang mit TCs Editor. Wer will, kann das Programm ja nach Belieben erweitern.

Die Projektdatei

Die für Jump benötigte Projektdatei finden Sie in Listing 3. Wenn Sie diese Datei benutzen, müssen Sie natürlich den C-Quelltext *TCJUMP.C* und das Assemblerlisting *TC-KUCK.S* nennen. Wenn Sie dann das Programm mit *Make TCJ.PRJ* compilieren und linken, wird es gleich als *TC-JUMP.ACC* (also mit dem richtigen Extender) auf Diskette abgelegt.

Ich habe Jump mit den Versionen 1.1 und 2.0 von TC getestet, und zwar auf einem alten 520+ mit TOS 1.0 (6.2.86). Seit Monaten läuft es (bis auf die Macke beim Kopieren von Zeilen) einwandfrei.

Eine Stack-Größe von 100 Bytes reicht für Jump aus. Falls Sie in Compiler und Linker alle Schalter außer „-G" im Compiler ungesetzt lassen, sollte das Accessory eine Größe von 4157 Bytes besitzen. Viel Spaß damit!

Literatur:

[1]: Jankowski, Reschke, Rabich: „ATARI ST Profibuch"; Sybex Verlag; Düsseldorf 1989.

[2]: Stefan Höhn: „Resource Formate"; ST-Computer 7/8 1990, Seite 97ff



```
  1:  /*
  2:  Jump! Marken im TURBO C Editor.
  3:  Geschrieben von Stefan Dreckmann mit TURBO C V2.0
  4:  und (ein bischen) MAS-68K V1.5
  5:  */
  6:
  7:  /*-----Includes-----------------*/
  8:  #include<aes.h>
  9:  #include<tos.h>
 10:  #include<string.h>
 11:  #include<stdlib.h>
 12:
 13:  /*-----Defines------------------*/
 14:  #define ACC_NAME "  Jump! "
 15:  #define AUS 0
 16:  #define EIN !AUS
 17:  #define TASTATUR 1
 18:  #define MY_EVENT MU_MESAG | MU_TIMER
 19:  #define EVNT_TIME 200
 20:  #define TC_NAME "TC      "
 21:  /*
 22:  TC_NAME (für appl_find()) MUSS! 8 Zeichen
 23:  haben! (Rest: Leerzeichen)
 24:  */
 25:
 26:  #define SCAN_F1 0x3B0000L
 27:  #define SCAN_F10 0x440000L
 28:  #define SCAN_SHIFT_F1 0x540000L
 29:  #define SCAN_SHIFT_F10 0x5D0000L
 30:  #define CONTROL 4
 31:  #define SHIFT 1
 32:  #define NOTSTART 14
 33:  #define CONTROL_L 0x260012L
 34:  #define RETURN 0x1C000DL
 35:  #define ESCAPE 0x01001BL
 36:  #define SHIFT_PFEIL_LINKS 0x4B0034L
 37:  #define CONTROL_Y 0x2C0019L
 38:  #define INSERT 0x520000L
 39:
 40:  #define PAUSE 300
 41:  #define PAEUSCHEN 50
 42:
 43:  #define MAX_GEDULD 10
 44:  #define MAXMARKE 10
 45:  #define RUECKSPRUNG MAXMARKE-1
 46:  #define BELEGT 1
 47:
 48:  #define TRAP_ZWO 34
 49:
 50:  /*-----globale Variablen----------*/
 51:  int accStatus=AUS;    /* Staus des Accessories */
 52:
 53:  IOREC* ioPtr;      /* Zeiger auf IOREC_Struktur */
 54:  long* bufferAdr; /* Anfang des Tastaturpuffers */
 55:
 56:  void (*trapZwoAdr)();    /* Originaladr.Trap #2 */
 57:  OBJECT** TCobjAdr;   /* Adresse 'Find Line' Box */
 58:  char* inputAdr;      /* Adresse Inputzeile */
 59:  int formDo;          /* Flag für form_do() */
 60:  int objcDraw;        /* Flag für object_draw() */
 61:
 62:
 63:  typedef struct
 64:    {
 65:    char string[10];   /* Platz für den Ziffern */
 66:    int belegt;        /* Belegungsflag */
 67:    }marken;
 68:  marken marke[MAXMARKE];/* ein Feld von Marken */
 69:
 70:  /*--Funktionsdeklarationen----------*/
 71:  extern void kuckma_rein(void);/* Assemblerteil */
 72:
 73:  void acc_init(void);
 74:  int alert(int button,int meldung);
 75:  void init_iorec(void);
 76:  void switch_acc(void);
 77:  int switch_on(void);
 78:  void switch_off(void);
 79:  void action_keybd(int lesemarke);
 80:  char* get_inputAdr(void);
 81:  void tastendruck(long code);
 82:  int call_formular(void);
 83:  void formular_entfernen(void);
 84:  void gehezu_marke(int nr,int flag);
 85:  void marke_merken(int nr);
 86:  void zeile_merken(void);
 87:  void marke_entfernen(void);
 88:  void zeile_anpassen(int nr);
 89:  void autoswitch(void);
 90:
 91:  /*-----Hauptprogramm---------------*/
 92:  main()
 93:    {
 94:    int dummy;
 95:    int event,alteLesemarke,keybdLesemarke;
 96:    int i;
 97:    int msgBuf[8];      /* der Messagepuffer */
 98:
 99:    acc_init();
100:    init_iorec();
101:
102:    for(i=0;i<MAXMARKE;i++)/* alle Marken... */
103:      marke[i].belegt= !BELEGT;/* ...unbelegt */
104:
105:    keybdLesemarke = alteLesemarke = ioPtr->ibufhd;
106:
107:    for(;;)    /* Endlosschleife, da Accessory */
108:      { /* Warte auf Message- oder Zeitereignis: */
109:      event=evnt_multi(MY_EVENT,0,0,0,0,0,0,0,0,0,
110:                       0,0,0,0,msgBuf,EVNT_TIME,0,
111:                       &dummy,&dummy,&dummy,&dummy,
112:                       &dummy,&dummy);
113:
114:      /* Tastendruck */
115:      if(accStatus && (keybdLesemarke =
116:                    ioPtr->ibufhd) != alteLesemarke)
117:        action_keybd(alteLesemarke =
118:                     keybdLesemarke);
119:      else if(event & MU_MESAG)     /* Nachricht */
120:        if(msgBuf[0] == AC_OPEN)/*ACC angeklickt */
121:          switch_acc(); /* ein- oder ausschalten */
```

→

```
122:       else if(msgBuf[0] == AC_CLOSE && accStatus)
123:         autoswitch();/* auf AC_CLOSE reagieren */
124:       }
125:     }
126:
127:   /*—Funktionsdefinitionen—————————*/
128:   /*—Accessory installieren—————————*/
129:   void acc_init(void)
130:     {
131:     extern _app;
132:     int ap_id;
133:
134:     if(_app)    /* Wenn Accessory, _app == 0 */
135:       {
136:       alert(1,0); /*soll als PRG gestartet werden*/
137:       exit(0);
138:       }
139:   /* beim AES anmelden und in Menüleiste eintr. */
140:     ap_id=appl_init();
141:     menu_register(ap_id,ACC_NAME);
142:     }
143:
144:
145:   /* Iorec initialisieren,Pufferanfang ermitteln */
146:   void init_iorec(void)
147:     {
148:     ioPtr=Iorec(TASTATUR);/* Zeiger auf Struktur */
149:     bufferAdr=ioPtr->ibuf;/* Zeiger auf Buffer  */
150:     }
151:
152:   /*—Meldungen mittels form_alert()———————*/
153:   int alert(int button,int meldung)
154:     {
155:     static char* meldungen[]=
156:       {
157:       „[3][Dieses Programm nur|als Accessory „
158:                                   „starten.][Ach so]",
159:       „[2][          Jump!    |hilft TC auf die „
160:                          „Sprünge][ Ein | Aus |Löschen]",
161:       „[3][Dieses Accessory nur|mit TURBO C „
162:                                   „benutzen!][Na gut]",
163:       „[3][Fehler beim Einschalten|von Jump!]"
164:                                             „[Mist!]",
165:       „[1][Diese Taste ist nicht belegt!]"
166:                                            „[Ach so!]",
167:       „[3][Dies ist die Rücksprungtaste!]"
168:                                   „[ Egal! |Abbruch]",
169:       };
170:
171:     return(form_alert(button,meldungen[meldung]));
172:     }
173:
174:   /*—Acc. ein- oder ausschalten——————————*/
175:   void switch_acc(void)
176:     {
177:     int i,but,antwort;
178:
179:     but = (accStatus == AUS) ? 1 : 2;
180:     if( (antwort=alert(but,1)) == 1)  /* ein? */
181:       if(appl_find(TC_NAME) >= 0)/* Hauptprogramm
182:                                             ist TC */
183:         if(switch_on() == 0) /* Einschalten
184:                                       erfolgreich? */
185:           accStatus=EIN;      /* Flag setzen */
186:         else
187:           alert(1,3);
188:       else
189:         {
190:         alert(1,2);
191:         accStatus=AUS;
192:         }
193:     else if(antwort==2)  /* ausschalten? */
194:       accStatus=AUS;      /* Flag löschen */
195:     else                  /* löschen */
196:       for(i=0;i<MAXMARKE;i++)
197:         marke[i].belegt = !BELEGT;
198:     }
199:
200:   /*—ACC einschalten————————————————*/
201:   int switch_on(void)
202:     {
203:     if(accStatus == AUS) /* ist es jetzt aus? */
204:       {
205:       if((inputAdr=get_inputAdr()) != 0)
206:         return 0;
207:       else
208:         return 1;
209:       }
210:     return 0;
211:     }
212:
213:   /*——auf Tastendruck reagieren————————*/
214:   void action_keybd(int lesemarke)
215:     {
216:     long code;
217:     int index;
218:     int flag = AUS;
219:
220:     code = *(bufferAdr + lesemarke/4); /* Code
221:                                              lesen */
222:     if(code >= SCAN_F1 && code <= SCAN_F10)
223:       {          /* Funktionstaste ohne SHIFT ? */
224:       accStatus=AUS;        /* Acc zeitweise aus */
225:       index=(int)(code >> 16) - 0x3B;/* Scancode in
226:                                   Feldindex umrechnen */
227:       if( marke[index].belegt )/* Taste belegt? */
228:         {
229:         if(Kbshift(-1) == CONTROL)
230:           flag = EIN;
231:         gehezu_marke(index,flag);/* Marke
232:                                        anspringen */
233:         }
234:       else
235:         alert(1,4);
236:       accStatus=EIN;          /* Acc wieder ein */
237:       }
238:     else if(code >= SCAN_SHIFT_F1 &&
239:            code <= SCAN_SHIFT_F10) /* mit SHIFT? */
240:       {
241:       accStatus=AUS;
242:       index=(int)(code >> 16) - 0x54;  /* Scancode
243:                              in Feldindex umrechnen */
244:       marke_merken(index);
245:       accStatus=EIN;
246:       }
247:     }
248:
249:   /*—Adresse der Eingabezeile ermitteln————*/
250:   char* get_inputAdr(void)
251:     {
252:     OBJECT* adr;
253:
254:     TCobjAdr=0;               /* Adresse auf 0 */
255:     if(call_formular())  /* Dialogbox aufrufen */
256:       {
257:       if(TCobjAdr == 0 || *TCobjAdr == 0)/* Fehler
258:                                 (z.B im Desktop) */
259:         {
260:         formular_entfernen();
261:         return 0;
262:         }
263:       adr= *TCobjAdr;       /* Adresse übernehmen */
264:
265:     /* das gesuchte Objekt muP „EDITABLE" sein: */
266:       while(adr->ob_flags != EDITABLE)
267:         {
268:         if(adr->ob_next >0) /* entweder nächstes
269:                                   Elternobjekt... */
270:           adr = *TCobjAdr + adr->ob_next;
271:         else if(adr->ob_head >0)    /* ...oder
272:                               nächstes Kindobjekt */
273:           adr = *TCobjAdr + adr->ob_head;
274:         else        /* falls keines mehr übrig... */
275:           {
276:           formular_entfernen();
277:           return 0;          /* ...Notfall!!! */
278:           }
279:         }
280:       formular_entfernen();
281:       marke_entfernen();
282:       return(adr->ob_spec.tedinfo->te_ptext);/*Adr.
283:                des Eingabestrings zurückliefern */
284:       }
285:     return 0;
286:     }
287:
288:   /*——'Find Line' Formular holen—————*/
289:   int call_formular(void)
290:     {
291:     int time_out=0;            /* „Rundenzähler" */
292:
293:     formDo=0;            /* form_do - Flag löschen */
294:     objcDraw=0;     /* object_draw - Flag löschen */
```

→

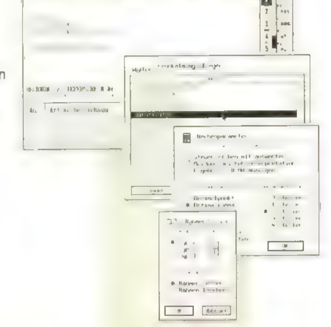
WRITER ST

Neu Version 2.0

WRITER ST wurde speziell für Personen entwickelt, die täglich eine große Anzahl an Briefen, Texten, Rechnungen oder kleineren Dokumentationen schreiben müssen, wie klein- und mittelständische Betriebe, Handwerker, Ärzte und Anwälte. Durch die konsequente Einbindung in die graphische Benutzeroberfläche GEM ist sie für den Einsteiger leicht und schnell zu erlernen.

- Die kommerzielle Textverarbeitung auf dem ATARI ST
- Rechnen und Fakturieren im Text
- integrierte Formularverwaltung
- Makroverwaltung mit bis zu 32.000 Makros (Artikel, Adressen...)
- Serienbriefschreibung (Mail-Merge) mit Schnittstelle zu Datenbanken
- vielfältige zeilen- und spaltenweise Blockoperationen
- bis zu 4 frei belegbare Tastaturen
- eigene Zeichensätze verwendbar
- lernfähiger Trennkatalog
- eigene Briefkopferstellung
- komfortable Druckeranpassung
- lauffähig auch auf Großbildschirmen
- und vieles, vieles mehr

komplett 189,-DM incl. Mwst.



SSD-SOFTWARE
M. Schmitt-Degenhardt
Gregorstr. 1 - D-5100 Aachen
Tel. 0241/602898

Schweiz: DTZ DataTrade AG - Landstr 1 - CH-5415 Rieden/Baden - Tel. 056/821880
Österreich: Haider Computer & Peripherie - Grazer Str 63 - A-2700 Wiener Neustadt - Tel. 02622/24280-0
Frankreich: LOG-ACCESS - 44 rue du Temple - F-75004 Paris - Tel. 42777456



SciGraph 2.0

BILD DER ZAHL

SciGraph 2.0 (AtariST/TT) kostet: 599 DM

eisindizes 1981 bis 1989

Bürogebäude	Landwirtsch.	Gewerbl.
106.1	105.8	106.1
110.0	108.9	110.4
112.6	111.1	112.9
115.8	113.8	115.7
116.8	114.3	116.5
118.9	116.9	118.9
21.5	118.1	121.5
24.6	120.5	124.1
90	124.8	128

1981 1982

da vinci design

SciLab GmbH
Isestraße 57
2000 Hamburg 13
Tel.: 040-4603702
Fax: 040- 479344

SciLab

```
295:    trapZwoAdr=Setexc(TRAP_ZWO,kuckma_rein);/* Ass.
296:                                 Routine einhängen */
297:   Kbshift(CONTROL); /* CONTROL */
298:    tastendruck(CONTROL_L);/* Tastendruck
299:                           CONTROL L simulieren */
300:    while( !objcDraw )/* warten bis TC reagiert */
301:      {
302:      evnt_timer(PAEUSCHEN,0);
303:      if(time_out++ >= MAX_GEDULD)/*klappt nicht*/
304:        {
305:        Setexc(TRAP_ZWO,trapZwoAdr);/* Original
306:                              routine einhängen */
307:        Kbshift(0);           /* normal */
308:        return 0;
309:        }
310:      }
311:    Kbshift(0);               /* Tastatur normal */
312:    while( !formDo )/* warten bis die Box fertig */
313:      evnt_timer(PAEUSCHEN,0);
314:    Setexc(TRAP_ZWO,trapZwoAdr); /* Original
315:                                routine einhängen */
316:
317:    return 1;
318:    }
319:
320:  /*—Formular entfernen————————*/
321:  void formular_entfernen(void)
322:    {
323:    tastendruck(RETURN); /* RETURN simulieren */
324:    }
325:
326:  /*-Simulation eines Tastendrucks—————*/
327:  void tastendruck(long code)
328:    {
329:    *(bufferAdr) = code; /* Code an den Anfang des
330:                          Tastaturpuffers schreiben */
331:    ioPtr->ibuftl=0;         /* Schreib- und */
332:    ioPtr->ibufhd=ioPtr->ibufsiz;  /* Lesemarke
333:                                       versetzen */
334:    while( ioPtr->ibufhd == ioPtr->ibufsiz )
335:      evnt_timer(PAEUSCHEN,0);/* warten bis Zeichen
336:                                      ausgelesen */
337:    }
338:
339:  /*—Marke anspringen————————*/
340:  void gehezu_marke(int nr,int flag)
341:    {
342:    char* position;
343:    long code;
344:    char rueck[10];
345:
346:    if(flag)              /* falls gewünscht... */
347:      zeile_merken();    /* akt. Zeile merken */
348:    if(call_formular()) /* Formular aufrufen */
349:      {
350:      if(nr != RUECKSPRUNG)/* es ist nicht die
351:                                Rücksprungtaste */
352:        {    /* Zeile für Rücksprung merken: */
353:        strcpy(marke[RUECKSPRUNG].string,inputAdr);
354:        marke[RUECKSPRUNG].belegt = BELEGT;/* Bele
355:                               gungsflag Rücksprung */
356:        }
357:      else /* Falls es die Rücksprungraste war */
358:        strcpy(rueck,inputAdr); /* Zeilennummer
359:                                 zwischenspeichern */
360:      position=marke[nr].string;/* Zeiger auf den
361:                                     Zeilenstring */
362:      tastendruck(ESCAPE); /* ESCAPE-Taste
363:                                       simulieren */
364:      while ((code=(long)*position++) != 0)
365:        tastendruck(code); /* Tastendruck auf
366:                            Zifferntaste simulieren */
367:      formular_entfernen();     /* bewirkt Sprung */
368:      marke_entfernen();
369:      if(flag)             /* falls Kopierfunktion */
370:       zeile_anpassen(nr);/* fertig zum Einfügen */
371:      if(nr == RUECKSPRUNG) /* Rücksprungtaste */
372:        strcpy(marke[RUECKSPRUNG].string,rueck);
373:                         /* gespeicherte Nr. holen */
374:      }
375:    }
376:
377:  /*—Marke festlegen————————*/
378:  void marke_merken(int nr)
379:    {
380:    if(nr == RUECKSPRUNG)/* Rücksprungtaste sollte
381:                            nicht belegt werden */
382:      if(alert(2,5) == 2)/* Warnung akzeptiert? */
383:        return;             /* keine Aktion */
384:
385:    if(call_formular())    /* Formular aufrufen */
386:      {
387:      strcpy(marke[nr].string,inputAdr);  /* Zeilen
388:                                string übernehmen */
389:      formular_entfernen();
390:      marke_entfernen();
391:      marke[nr].belegt = BELEGT;/* Belegungsflag */
392:      }
393:    }
394:
395:  /*——Zeile in den Zwischenspeicher———*/
396:  void zeile_merken(void)
397:    {
398:    Kbshift(CONTROL); /* CONTROL */
399:    tastendruck(CONTROL_Y);/* Zeile löschen
400:                           (kommmt in den Buffer) */
401:    evnt_timer(PAUSE,0);    /* Reaktion abwarten */
402:    tastendruck(INSERT);/* Zeile wieder einfügen */
403:    evnt_timer(PAUSE,0);  /* wieder warten */
404:    Kbshift(0);           /* Tastatur normal */
405:    }
406:
407:  /*———Marke entfernen————————*/
408:  void marke_entfernen(void)
409:    {
410:    Kbshift(SHIFT);            /* SHIFT */
411:    tastendruck(SHIFT_PFEIL_LINKS);   /* Cursor an
412:                           den Anfang der Zeile...*/
413:    Kbshift(0);                /* Tastatur normal */
414:    }
415:
416:
417:  /*————Zeilennummer erhöhen—————*/
418:  void zeile_anpassen(int nr)
419:    {
420:    int akt_zeile;
421:    int zeile,i;
422:
423:    akt_zeile=atoi(marke[nr].string);
424:    for(i=0;i < MAXMARKE;i++)
425:      if((zeile=atoi(marke[i].string))>=akt_zeile)
426:        itoa(++zeile,marke[i].string,10);
427:    }
428:
429:  /*—automatisches Ein-/Ausschalten————*/
430:  void autoswitch(void)
431:    {
432:    int msg[8];
433:    int dummy;
434:
435:    for(;;)
436:      {
437:      evnt_mesag(msg);  /* auf Nachricht warten */
438:      if(msg[0] == AC_CLOSE)/* Neustart des ACCs */
439:        {
440:        evnt_multi(MY_EVENT,0,0,0,0,0,0,0,0,0,0,
441:                   0,0,msg,PAUSE,0,&dummy,&dummy,
442:                   &dummy,&dummy,&dummy,&dummy);
443:        if(appl_find(TC_NAME) < 0)/* neues Haupt
444:                           programm ist nicht TC */
445:          accStatus = AUS;     /* ACC ausschalten */
446:        break;              /* Schleife abbrechen */
447:        }
448:      else if(msg[0] == AC_OPEN)      /* ACC soll
449:                                 gestartet werden */
450:        if(Kbshift(-1) == NOTSTART)  /* aber nur
451:                                     per Notstart */
452:          {
453:          accStatus = AUS;   /* Flag auf Aus */
454:          switch_acc();      /* Startroutine */
455:          break;            /* Schleife verlassen */
456:          }
457:      }
458:    }
```

Listing 1: Das Hauptprogramm...

```
 1:  ;die Assemblerroutine kuckma_rein() in Trap #2
 2:
 3:  GLOBL kuckma_rein ;Einsprungadresse der Funktion
 4:  GLOBL trapZwoAdr  ;alte Einsprungadresse Trap #2
 5:  GLOBL formDo      ;Flag form_do
 6:  GLOBL objcDraw    ;Flag objc_draw
 7:  GLOBL TCobjAdr    ;für Zeiger auf Adresse des
 8:                                    ;Objektbaums
 9:
10:  vdi EQU 115       ;Opcode VDI
11:  obdra EQU 42      ;Opcode objc_draw()
12:  fodo EQU 50       ;Opcode form_do()
13:
14:  kuckma_rein:
15:    cmpi.w #vdi,d0         ;falls VDI Routine...
16:    beq ausgang            ;...nichts machen
17:
18:    move.l d1,a0           ;Adresse AESPB holen
19:    move.l (a0),a1         ;Adresse CONTROL-Feld
20:                                               ;holen
21:
22:    cmpi.w #obdra,(a1)     ;Opcode für objc_draw()?
23:    bne next               ;nein: keine Aktion
24:    addq.w #1,objcDraw     ;Flag setzen
25:    move.l 16(a0),TCobjAdr ;Zeiger auf Adresse des
26:                              ;Objekbaums übergeben
27:
28:  next:
29:    cmpi.w #fodo,(a1)      ;Opcode für form_do()?
30:    bne ausgang            ;nein: keine Aktion
31:    addq.w #1,formDo       ;Flag setzen
32:
33:  ausgang:
34:    move.l trapZwoAdr,a0   ;alte Trap-Adresse
35:                                               ;holen
36:    jmp   (a0)             ;dort fortfahren
```

Listing 2: ...und der Assembler-Teil

DEMO DISKS

Demo-Disketten

Damit Sie nicht immer die Katze im Sack kaufen müssen, haben wir ab sofort eine neue Rubrik für Sie eingeführt; es sind Demo-Disketten kommerzieller Software. Sie kosten lediglich DM 10,- pro Diskette und können über die Redaktion bezogen werden. So müssen Sie zum Vergleich verschiedener Programme nicht an verschiedene Hersteller schreiben, sondern können sich in aller Ruhe das Demonstrationsprogramm ansehen, bevor Sie das Original kaufen.

Bitte beachten Sie, daß die angebotenen Disketten nur Demonstrationsdisketten der Originalversionen sind und somit im Gegensatz zu den Originalen in Funktion eingeschränkt sind!

Folgende Demo-Disketten sind z.Zt. erhältlich:

D1: S.&P.-Charts
Chart-Analyseprogramm
(*S.P.S. Software*)

D2: SPC-Modula-2
Modula-2-Entwicklungssystem
(*Advanced Applications Viczena*)

D3: ST-Fibu
Finanzbuchhaltungsprogramm
(*GMa-Soft*)

D4: ST-Fibu-Fakt
Fakturierungsprogramm für ST-Fibu
(*GMa-Soft*)

D5: ST-Fibu-Text
Textverarbeitungsprogramm für ST-Fibu mit Serienbrieffunktion
(*GMa-Soft*)

D6: SciGraph
Programm zur Erstellung von Präsentationsgrafiken
(*SciLab GmbH*)

D7: ST-Statistik
Uni- und multivariates Statistikprogramm, Grafikeinbindung (*SciLab GmbH*)

D8: fibuSTAT
Finanzbuchhaltungs-/Statistikprogramm
(*novoPLAN Software GmbH*)

D9: Btx/Vtx-Manager
Programm zum Anschluß an Bildschirmtext
(*Drews Btx + EDV GmbH*)

D10: Edison
Editor für fast alle Gelegenheiten
(*Kniss Soft*)

D11 & D12: CADjA
CAD-Programm für hohe Ansprüche
(*Computer Technik Kieckbusch*).
Demo besteht aus zwei Disketten zu je DM 10,-!

D13: JAMES 2.0
Programm für Börsenspekulanten
(*IFA-Köln*)

D14: Soundmerlin
Sample-Editor-Programm mit vielen Modulen
(*TommySoftware*)

D15: Soundmachine II
Programm zur Erstellung und Wiedergabe von Sounds
(*TommySoftware*)

D16: ReProK
Büroorganisationsprogramm
(*Stage Microsystems*)

D17: Sherlook
Schrifterkennungs- und -verarbeitungsprogramm
(*H.Richter*)

D18: ST Matlab
Programmiersystem mit Schnittstelle zu Modula-2
(*Advanced Aplications Viczena*)

D19: Calamus
Desktop-Publishing-Programm
(*DMC*)

D20: GD-Fibu
Finanzbuchhaltungsprogramm
(*GDAT*)

D21: Omikron.Draw!
Zeichen- und Malprogramm
(*Omikron.Software*)

D22: Omikron.Libraries
Verschiedene Libraries für Omikron.BASIC
(*Omikron.Software*)

D23: Omikron.Compiler
Demo-Version des Omikron.BASIC-Compilers
(*Omikron.Software*)

D24: Mortimer
Multi-Programm für alle Gelegenheiten
(*Omikron.Software*)

D25: Script 1
Textverarbeitungsprogramm
(*Application Systems /// Heidelberg*)

D26: SuperScore
Sequencer- und Notendruckprogramm
(*BELA Computer GmbH*)

D27: SPS-Emulator: nicht mehr lieferbar

D28: STAD 1.3+
Zeichenprogramm mit 3D-Teil
(*Application Systems /// Heidelberg*)

D29: MegaFakt
Fakturierungsprogramm
(*MegaTeam*)

D30 & D31: MegaPaint II
Zeichenprogramm mit Vektorteil
(*TommySoftware*)

D32: Tempus Word
Textverarbeitung
(*CCD*)

D33: Creator
Zeichenprogramm mit Animationsteil
(*Application Systems /// Heidelberg*)

D34: Outline Art
Utility für Calamus
(*DMC*)

D35: compugraphic Schriften
für Calamus
(*DMC*)

D36: BTX-Börsen-Manager
Börsenprogramm
(*Thomas Bopp Softwarevertrieb*)

D37: Cashflow
Kassenbuch
(*C.A.$.H.*)

D38: TiM II
Finanzbuchhaltungsprogramm
(*C.A.$.H.*)

D39: nicht mehr lieferbar

D40: Technobox Drafter
(Zeichenprogramm spez. f. Konstruktionen)
(*Technobox*)

D41: Platon
(Leiterplatten- CAD-System)
(*VHF-Computer*)

D42: Script 2
Textverarbeitungsprogramm
(*Application Systems /// Heidelberg*)

Es gelten die gleichen Vertriebsbedingungen wie für PD-Disketten (s. PD-Seiten am Ende dieser Ausgabe). Demo-Disketten können auch zusammen mit PD- und Sonder-Disketten bestellt werden.

Bitte vergessen Sie nicht die betreffende Bestellnummer (z.B. D1) anzugeben.

GEDREHTE BILDSCHIRM-AUSSCHNITTE

Philip Köster

GFA-PROGRAMMIERER WISSEN DIE EFFEKTE DES ALLMÄCHTIGEN PUT-BEFEHLS ZU SCHÄTZEN. WIE WÄRE ES ABER, WENN MAN IHM NOCH EINIGE WEITERE PARAMETER MIT AUF DEN WEG GEBEN KÖNNTE, ETWA FÜR EINEN GENAUEN DREHWINKEL ODER DIE GEWÜNSCHTE ABBILDUNGSGRÖSSE ...?

Diese kleine Routine gestattet es, einen per GET eingelesenen Ausschnitt in einem beliebigen Winkel zu drehen und in ebenso beliebiger Größe darzustellen. Sie arbeitet nur in hoher Auflösung (640 x 400 Punkte). Die Breite des Ausschnitts muß ganzzahlig durch 16 teilbar sein, die Wahl der Höhe allerdings ist frei. Beim Drehen stehen die vier gebräuchlichsten Grafikmodi zur Verfügung.

Die Praxis

Die Syntax der Prozedur ist eng an den PUT-Befehl angelehnt und von daher denkbar einfach: *put(x,y,b,h,a$,m,a)*. *x* und *y* beschreiben dabei - wie gehabt - die Koordinate der linken oberen Ecke des (eventuell) gedrehten Ausschnitts, *b* und *h* die gewünschte Breite und Höhe. Für *b* und *h* gilt: Ein negativer Wert zeigt an, wie die Originalgröße dargestellt werden soll, ein Mitführen von Variablen zur Speicherung dieser Werte ist also nicht erforderlich. *a$* hat den Grafikausschnitt zum Inhalt und ist durch GET zu initialisieren. *m* zeigt den Modus der Verknüpfung von Vorder- und Hintergrund an; möglich sind die Werte von 1 bis 4 (s. u.). *a* ist schließlich der Rotationswinkel in Altgrad (Null bis 360, gültig sind natürlich auch alle Kommazahlen), um den der Ausschnitt im Uhrzeigersinn gedreht werden soll (s. Abb. 1).

Die Prozedur arbeitet ohne jeden Seiteneffekt; sie verändert also weder die Übergabevariablen noch irgendwelche Grafikeinstellungen (auch nicht den Bildschirmmodus). Beim Einsatz dieser Routine reagiert die Grafikausgabe auch auf Clipping - im Gegensatz zum konventionellen PUT-Befehl. Leider ergeben sich in der Anwendung dieser Prozedur einige Einschränkungen:

1. Die Projektion eines 96 x 96 Punkte großen Ausschnitts nimmt im Schnitt etwa 45 Sekunden in Anspruch, das Kompilat benötigt immerhin noch 35. Mit einem erweiterten PUT-Befehl haben wir es also nur bedingt zu tun. Diese Routine wird ihren Einsatz also »nur« zum Zwecke von Bildgestaltungen (z. B. in Zeichenprogrammen) finden, nicht aber zur Realisierung von Animationen.
2. Beim Drehen kann der Bildschirm, der ja den Aufbau einer Matrix hat, nicht »flächendeckend« bemalt werden. So bleibt ein gedrehtes schwarzes Rechteck nicht überall schwarz, und ein weißes Rechteck, projiziert auf einen dunklen Hintergrund, ist an einigen Stellen zwangsläufig gesprenkelt. Diese unliebsamen Effekte sind nicht auf eine Unzulänglichkeit des Programms zurückzuführen, sondern rühren von der Natur der Rastergrafiken her. Hier hilft nur die Korrektur durch ein Zeichenprogramm. Soll ein Ausschnitt auf einem durchweg weißen Hintergrund abgebildet werden, ist Modus 2 (durchsichtig) dem Modus 1 (ersetzen) in jedem Fall vorzuziehen.
3. Die Veränderung der Ausschnittgröße klappt zwar tadellos, jedoch entstehen bei der Vergrößerung Lücken in dunklen Flächen (wieder dasselbe Problem!). Hier ließe sich die Routine übrigens noch verbessern, indem sie bei der Vergrößerung weiß oder schwarz gefüllte Rechtecke statt kleiner Punkte zeichnete (was allerdings hübsche »Treppen« entstehen läßt). Ohnehin bietet es sich aber an, Ausschnitte, die größenverändert dargestellt werden sollen, anfangs möglichst groß zu entwerfen und hinterher zu verkleinern.
4. Bei Rotationswinkeln, die nicht ein Vielfaches von 90 sind, kann ein Ausschnitt in seiner Größe nicht unproportional verändert werden; in solchen Fällen muß also beispielsweise ein Quadrat ein Quadrat bleiben - dies allerdings in beliebiger Größe. Im Bedarfsfall wird der Parameter *h* automatisch entsprechend korrigiert.

Die Theorie

Zur mathematischen Betrachtung der Drehung: Gegeben sei eine Gerade $\overline{AB}$ mit $A = \{3;2\}$ und $B = \{8;2\}$ (s. Abb. 2). Soll diese Gerade am Punkt A um 10 Altgrad im Uhrzeigersinn gedreht werden, berechnet sich der Punkt B' wie folgt (alte Schneider-Freaks werden sich erinnern):

$$B' = \{3 + (8 - 3) \cos 10;2 + (8 - 3) \sin 10\} \approx \{7{,}92;2{,}87\}.$$

Einen Beweis bleibe ich an dieser Stelle der Bequemlichkeit halber schuldig.

Mit diesem Wissen läßt sich nun der Inhalt eines Rechteckes der Breite *b* und der Höhe *h* drehen. Δy läuft dabei in Einerschritten die Werte von 0 bis *h* durch, Δx analog von 0 bis *b*. So wird jeder Punkt $\{\Delta x;\Delta y\}$ des Rechteckes abgefragt, in meiner Routine zeilenweise von oben nach unten, in der Zeile von links nach rechts. Wie sich der durch GET erhaltene String zu diesem Zweck zerpflücken läßt, ist weiter unten beschrieben.

Nun findet der »zweifache Schneider« Einsatz (s. Abb. 3):
1. Zunächst müssen von jeder gedrehten Zeile die Koordinaten des Rotationspunktes $R = \{x_0;y_0\}$, der gleichzeitig den Anfangspunkt darstellt, ermittelt werden. Nichts leichter als das: $R = \{x + \Delta y \cos(90 + \alpha);y + \Delta y \sin(90 + \alpha)\}$. $\{x;y\}$ ist dabei der linke obere Punkt des gedrehten Rechteckes, α der Rotationswinkel.
2. Nun finden sich die Koordinaten eines jeden Punktes im gedrehten Rechteck: $\{\Delta x;\Delta y\}' = \{x_0 + \Delta x \cos \alpha;y_0 + \Delta x \sin \alpha\}$. Das war's.

Das Programm

Im Programm werden die GFA-Funktionen SINQ und COSQ verwendet. Dies hat den praktischen Grund, daß diese - im Gegensatz zu SIN und COS - gleich in den gewünschten Altgrad rechnen. Zudem sind die verwendeten Funktionen etwas schneller; eine etwaige Rechenungenauigkeit hat für den Einsatz der Berechnung von Bildschirmkoordinaten keinerlei Bedeutung.

Der Befehl GET liefert einen String zurück, der folgenden Aufbau hat: a$ = MKI$(b - 1) + MKI$(h - 1) + MKI$(1) (konstant) + MKI$(1. 16 Punkte) + MKI$(2. 16 Punkte) + ... Jeder MKI$ ist zwei Zeichen lang und kann alle Werte von Null bis 65535 enthalten. Dieser Aufbau gestaltet sich etwas komplizierter, wenn *b* nicht ganzzahlig durch 16 teilbar ist oder im Farbbetrieb gearbeitet wird.

Per CVI erhält man den Wert eines einzelnen MKI$s, und ein

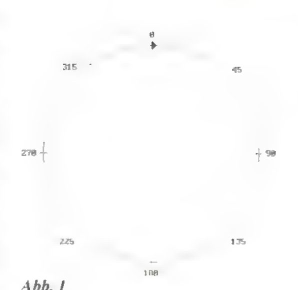


Abb. 1

einzelnes Bit a kann via BTST isoliert werden. Wer jetzt noch die genaue Formel wissen möchte, möge sich den Bandwurm im Listing zu Gemüte führen.

Nun zur Verknüpfung des Inhalts des gedrehten Rechteckes und des Hintergrundes, auf den dieses projiziert werden soll. Das Bit *b*, das die Information enthält, ob ein Punkt auf dem Bildschirm an der entsprechenden Koordinate bereits gesetzt ist oder nicht, kann durch POINT initialisiert werden. *c* sei schließlich das Bit, das aus der logischen Verknüpfung von *a* (Punkt im Rechteck) und *b* hervorgeht. Abhängig von *c* wird schließlich die Farbe des gesetzten Punktes gewählt (FALSE: 0, TRUE: 1).

Modus	Verknüpfung	Effekt
1	c = a	Ersetzen
2	c = a OR b	durchsichtig
3	c = a XOR b	negativ
4	c = (NOT a) OR b	durchsichtig und negativ

Weitere Bildschirmmodi (es gibt überhaupt nur 16 mögliche) können schnell realisiert werden; die Routine ist in diesem Punkt sehr leicht zu verstehen.

Das Programm demonstriert die Arbeitsweise der Prozedur *put* (s. Abb. 4). Im linken oberen Viertel des Bildschirms zeigt es zwei Beispiele für eine Hintergrundverknüpfung. Rechts daneben wird der Ausschnitt größenverändert dargestellt. In der unteren Bildschirmhälfte sind schließlich einige Drehbeispiele zu bewundern.

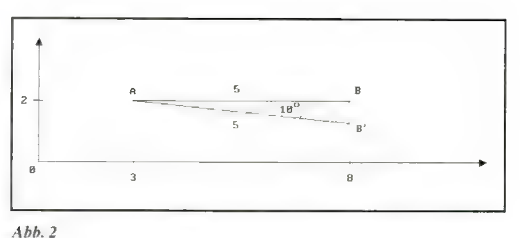


Abb. 2

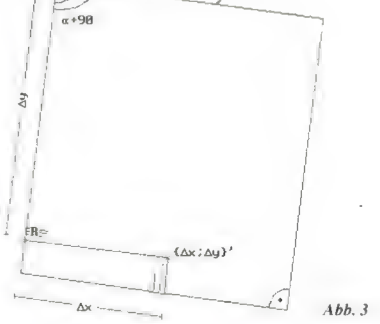


Abb. 3

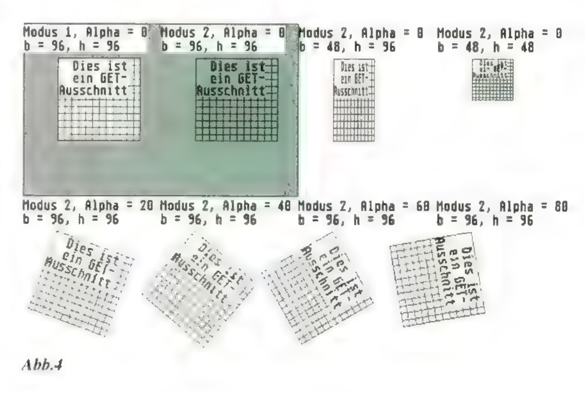


Abb.4

```
 1:  ' Gedrehte Bildschirmausschnitte in GFA-BASIC 3.0
 2:  ' (c) 1990 by Philip Köster
 3:  ' Am Waldbad 13 - 2122 Bleckede 6
 4:  '
 5:  DEFFILL 1,3,12              !Ausschnitt zeichnen
 6:  PBOX 0,0,95,95
 7:  '
 8:  TEXT 1,14,"  Dies ist"
 9:  TEXT 1,28,"  ein GET-"
10:  TEXT 1,42,"Ausschnitt"
11:  '
12:  GET 0,0,95,95,ausschnitt$   !... und speichern
13:  '
14:  DEFFILL 1,2,4  !Demo: Hintergrundverknüpfung
15:  PBOX 0,0,319,199
16:  '
17:  FOR i|=1 TO 2
18:    TEXT (i|-1)*160,15,"Modus „+STR$(i|)+", Alpha
       = 0"
19:    TEXT (i|-1)*160,31,"b = 96, h = 96    „
20:    '
21:    put(40+(i|-1)*160,40,-1,-1,ausschnitt$,i|,0)
                                       !Prozedur-Aufruf
22:  NEXT i|
23:  '
```
→

```
24:  TEXT 320,15,"Modus 2, Alpha = 0"  !Demo:
                                     Größenveränderung
25:  TEXT 320,31,"b = 48, h = 96"
26:  '
27:  put(360,40,48,96,ausschnitt$,2,0)
28:  '
29:  TEXT 480,15,"Modus 2, Alpha = 0"
30:  TEXT 480,31,"b = 48, h = 48"
31:  '
32:  put(520,40,48,48,ausschnitt$,2,0)
33:  '
34:  FOR i|=1 TO 4  !Demo: Drehen
35:    TEXT (i|-1)*160,215,"Modus 2, Alpha = „+
                        STR$(i|*20)
36:    TEXT (i|-1)*160,231,"b = 96, h = 96"
37:    '
38:    put(40+(i|-1)*160,240,-1,-1,ausschnitt$,2,i|*
                                    20)
39:  NEXT i|
40:  '
41:  PRINT AT(1,25);"Taste drücken.";
42:  ~INP(2)
43:  '
44:  END  !Jetzt geht's erst los ...
45:  '
46:  PROCEDURE put(x&,y&,pb&,ph&,ausschnitt$,modus|,
                 alpha&)
47:    LOCAL dx&,dy&,x0&,y0&,b&,h&,fx,fy,quelle!,
        ziel!,ergebnis!  !Nicht im Hauptprg stören
48:    '
49:    b&=CVI(MID$(ausschnitt$,1,2))+1  !Breite ...
50:    h&=CVI(MID$(ausschnitt$,3,2))+1  !... und Höhe
                           des Ausschnitts feststellen
51:    '
52:    IF pb&<0  !Keine Größenveränderung in x-...
53:      pb&=b&
54:    ENDIF
55:    IF ph&<0  !... oder y-Richtung?
56:      ph&=h&
57:    ENDIF
58:    '
59:    fx=pb&/b&  !Projektionsfaktoren in x-...
60:    fy=ph&/h&  !... und y-Richtung feststellen
61:    '
62:    IF alpha& MOD 90  !"Ungerade Drehung": Keine
                 Veränderung der Proportionen erlaubt
63:      fy=fx
64:    ENDIF
65:    '
66:    FOR dy&=0 TO h&-1  !Von oben nach unten
67:      x0&=x&+dy&*fx*COSQ(90+alpha&)
                !Rotationspunkt der aktuellen ...
68:      y0&=y&+dy&*fy*SINQ(90+alpha&)  !...
       gedrehten Zeile feststellen
69:      '
70:      FOR dx&=0 TO b&-1  !Von links nach rechts
71:        quelle!=BTST(CVI(MID$(ausschnitt$,7+(dy&*
          b&/16+dx& DIV 16)*2,2)),15-dx& MOD
            16)  !Aktuelles Bit aus String isolieren
72:        '
73:        ziel!=POINT(x0&+dx&*fx*COSQ(alpha&),y0&+
         dx&*fy*SINQ(alpha&))  !Punkt gesetzt?
74:        '
75:        SELECT modus|  !Verknüpfungsart feststellen
76:        CASE 1  !GRAPHMODE 1
77:          ergebnis!=quelle!
78:        CASE 2  !GRAPHMODE 2
79:          ergebnis!=quelle! OR ziel!
80:        CASE 3  !GRAPHMODE 3
81:          ergebnis!=quelle! XOR ziel!
82:        CASE 4  !GRAPHMODE 4
83:          ergebnis!=(NOT quelle!) OR ziel!
84:        ENDSELECT
85:        '
86:        IF ergebnis!  !Farbe bestimmen
87:          COLOR 1
88:        ELSE
89:          COLOR 0
90:        ENDIF
91:        '
92:        PLOT x0&+dx&*fx*COSQ(alpha&),y0&+dx&*fy*
              SINQ(alpha&)  !Punkt setzen
93:      NEXT dx&
94:    NEXT dy&
95:  RETURN  !War doch gar nicht so viel.
```

DER ETWAS ANDERE ZUGRIFF

Uwe Seimet

In einigen Fällen kann es sinnvoll sein, aus eigenen Programmen heraus auf die Daten der Desktop-Info zuzugreifen. Wie man dies bewerkstelligen und welche Effekte man so erzielen kann, soll uns an dieser Stelle beschäftigen.

Befindet sich beim Bootvorgang eine Datei mit dem Namen "DESKTOP.INF" auf dem Bootlaufwerk, werden die Daten in dieser Datei vom GEM dazu verwendet, das Aussehen des Desktops und diverse Voreinstellungen festzulegen. Diese Daten bleiben solange gültig, bis sie vom Anwender verändert werden oder ein Reset ausgelöst wird. Welche Informationen die "DESKTOP.INF"-Datei enthält und wie diese organisiert sind, wurde bereits von Oliver Dietz in der ST-Computer 10/86 ausführlich behandelt. Da sich der Leserkreis inzwischen sicher vergrößert hat, möchte ich das Wichtigste hier noch einmal zusammenfassen. Anschließend sehen wir uns an, wie man diese Informationen vom GEM erfragen und verändern kann.

Die Datei "DESKTOP.INF" enthält Angaben über die RS232-Parameter, Druckerkonfiguration, Bildschirmauflösung und -farben, Kontrollfeldparameter, Namen der angemeldeten Anwendungen sowie Fenster- und Icon-Positionen. Diejenigen Parameter, die die Schnittstellen sowie Auflösung und Farben betreffen, können über diverse Betriebssystemaufrufe ermittelt und verändert werden. Die restlichen Angaben, die das GEM betreffen, sind jedoch über das Betriebssytem nicht ohne weiteres erreichbar. Es gibt jedoch Situationen, in denen es nötig ist, auf das Aussehen des Desktops Einfluß zu nehmen, bevor man durch das Beenden eines Programms in dieses zurückkehrt. So kann es eine Festplatte übelnehmen, wenn das Desktop versucht, ein Festplatten-Window zu öffnen, nachdem die Platte geparkt worden ist. Aus diesem Grund kann man solche Parkprogramme nicht von der Festplatte starten, wenn man nach dem Parkvorgang wieder zum Desktop zurückkehren möchte. Schließlich muß man in diesem Fall ein Festplatten-Window geöffnet haben, um das Parkprogramm zu starten. Das heißt jedoch, daß das Desktop versucht, dieses Window nach dem Verlassen des Programms erneut zu öffnen. Wie man dieses Problem lösen kann, soll Gegenstand der nächsten Zeilen sein. Ein kleines Assembler-Programm mit dem Namen *CLOSE* soll die Vorgehensweise beim Manipulieren der Desktop-Daten verdeutlichen und als Anschauungsmaterial für eigene Anwendungen dienen.

Die Daten der "DESKTOP.INF"-Datei werden nach einem Systemstart nicht nur vom GEM interpretiert, sondern die Datei wird in einem speziellen AES-Puffer gespeichert. Jede Änderung auf dem Desktop, die vom Anwender durchgeführt wird, wird sofort in diesem Puffer vermerkt. Der Menüpunkt *Arbeit sichern* schreibt den Inhalt des Puffers auf die Disk zurück. Ändern Sie die Anga-

```
 1:  GEMDOS   = 1
 2:  SETBLOCK= $4a
 3:
 4:
 5:          text
 6:
 7:          move.l sp,a0
 8:          lea stack+200,sp             ;Stackpointer
                                           initialisieren
 9:          move.l 4(a0),a0
10:          move.l 12(a0),a1
11:          add.l 20(a0),a1
12:          add.l 28(a0),a1
13:          lea $100(a1),a1
14:          move.l a1,-(sp)
15:          move.l a0,-(sp)
16:          clr -(sp)
17:          move #SETBLOCK,-(sp)
18:          trap #GEMDOS                 ;überzähligen
                                           Speicher frei
19:          lea 12(sp),sp
20:          tst.l d0                     ;alles klar?
21:          bne error                    ;leider nein-
22:
23:          lea intin,a5                 ;Pointer auf
                                           INTIN-Array
24:          lea intout,a6                ;Pointer auf
                                           INTOUT-Array
```

```
25:          moveq #10,d0                 ;appl_init
26:          move.l #$00010000,d1
27:          bsr.s aes
28:          tst (a6)                     ;Fehler?
29:          bmi.s error                  ;ja-
30:
31:          lea deskinf,a4               ;Puffer für
                                           Desktop-Info
32:          moveq #122,d0                ;shel_get
33:          move.l #$01010100,d1
34:          move #1024,(a5)              ;1024 Bytes
                                           kopieren
35:          move.l a4,addrin             ;Pufferadresse
                                           eintragen
36:          bsr.s aes                    ;AES-Puffer
                                           auslesen
37:
38:  loop:   move.b (a4)+,d0
39:          beq.s einfo
40:          cmp.b #26,d0                 ;CTRL-Z?
41:          beq.s einfo                  ;ja-
42:          cmp.b #'#',d0
43:          bne loop
44:          cmp.b #'W',(a4)              ;folgen
                                           Fensterdaten?
45:          bne loop                     ;nein-
46:          lea 23(a4),a4                ;Pointer auf
                                           Fensternamen →
```

gesamten 1024 Bytes zu kopieren, da ja im voraus nicht bekannt ist, wie umfangreich die Daten der Desktop-Info sind.

Nachdem die Daten erfolgreich im eigenen Puffer gelandet sind, können sie nach Herzenslust manipuliert werden. Im einfachsten Fall liest man nur ein paar Informationen aus, z.B. um festzustellen, welche Voreinstellungen über das Desktop vorgenommen wurden. Solche Angaben können auch für eigene Programme relevant sein. Denkbar wäre auch, die Daten zu erweitern, um eine weitere Anwendung anzumelden. In unserem Fall geht es jedoch darum, die Fensterdaten zu ändern. Zielsetzung soll es sein, alle Windows bis auf die der Laufwerke A und B zu schließen. So kann man sichergehen, daß bei der Rückkehr ins Desktop die Festplatte nicht mehr angesprochen wird. Gleiches gilt dann allerdings auch für RAM-Disks, da diese die gleichen Laufwerkskennungen wie Festplatten-Partitionen besitzen können. Die Unterscheidung zwischen Festplatten und RAM-Disks ist zwar programmtechnisch möglich, würde hier jedoch den Rahmen sprengen. Zurück also zur eigentlichen Aufgabe. Um die gewünschten Windows zu schließen, muß geprüft werden, ob die offenen Windows für die Floppies oder für die Festplatte zuständig sind. Da der erste Buchstabe des Window-Namens die Laufwerkskennung darstellt, ist dies keine Kunst. Handelt es sich um ein *A* oder ein *B*, braucht das zugehörige Window nicht geschlossen zu werden. Andernfalls wird ein “@” an die erste Stelle des Namens geschrieben. Wie bereits erwähnt, gelten solche Windows für das Desktop als geschlossen.

Hat man die gesamten Desktop-Daten, die mit einem CTRL-Z abgeschlossen sind, durchgearbeitet, müssen diese wieder in den AES-Puffer geschrieben werden. Hierzu bedient man sich des *SHEL_PUT*-Aufrufs, der die gleichen Parameter erhält wie *SHEL_GET*. Nachdem die Daten zurückgeschrieben worden sind, kann man getrost in Desktop zurückkehren. Die einzigen Windows, die nun geöffnet werden, beziehen sich auf die Floppies A und B, und die Festplatte hat ihre Ruhe.



```
  1:  GEMDOS   = 1
  2:  SETBLOCK= $4a
  3:
  4:
  5:          text
  6:
  7:          move.l sp,a0
  8:          lea stack+200,sp        ;Stackpointer
                                       initialisieren
  9:          move.l 4(a0),a0
 10:          move.l 12(a0),a1
 11:          add.l 20(a0),a1
 12:          add.l 28(a0),a1
 13:          lea $100(a1),a1
 14:          move.l a1,-(sp)
 15:          move.l a0,-(sp)
 16:          clr -(sp)
 17:          move #SETBLOCK,-(sp)
 18:          trap #GEMDOS            ;überzähligen
                                       Speicher frei
 19:          lea 12(sp),sp
 20:          tst.l d0                ;alles klar?
 21:          bne error               ;leider nein-
 22:
 23:          lea intin,a5            ;Pointer auf
                                       INTIN-Array
 24:          lea intout,a6           ;Pointer auf
                                       INTOUT-Array
 25:          moveq #10,d0            ;appl_init
 26:          move.l #$00010000,d1
 27:          bsr.s aes
 28:          tst (a6)                ;Fehler?
 29:          bmi.s error             ;ja-
 30:
 31:          lea deskinf,a4          ;Puffer für
                                       Desktop-Info
 32:          moveq #122,d0           ;shel_get
 33:          move.l #$01010100,d1
 34:          move #1024,(a5)         ;1024 Bytes
                                       kopieren
 35:          move.l a4,addrin        ;Pufferadresse
                                       eintragen
 36:          bsr.s aes               ;AES-Puffer
                                       auslesen
 37:
 38:  loop:   move.b (a4)+,d0
 39:          beq.s einfo
 40:          cmp.b #26,d0            ;CTRL-Z?
 41:          beq.s einfo             ;ja-
 42:          cmp.b #'#',d0
 43:          bne loop
 44:          cmp.b #'W',(a4)         ;folgen
                                       Fensterdaten?
 45:          bne loop                ;nein-
 46:          lea 23(a4),a4           ;Pointer auf
                                       Fensternamen
 47:          move.b (a4),d0
 48:          cmp.b #'@',d0           ;Fenster offen?
 49:          beq loop                ;nein-
 50:          cmp.b #'C',d0           ;Floppy-Fenster?
 51:          bcs loop                ;ja-
 52:          move.b #'@',(a4)        ;Fenster
                                       desaktivieren
 53:          bra loop
 54:
 55:  einfo:  moveq #123,d0           ;shel_put
 56:          move.l #$01010100,d1
 57:          move #1024,(a5)         ;1024 Bytes
                                       schreiben
 58:          move.l #deskinf,addrin
 59:          bsr.s aes               ;neuen Pufferin-
                                       halt zurück
 60:
 61:          moveq #19,d0            ;appl_exit
 62:          move.l #$00010000,d1
 63:          bsr.s aes
 64:
 65:  error:  clr -(sp)
 66:          trap #GEMDOS            ;das war's
 67:
 68:  *AES-Aufruf
 69:  *Funktionsnummer in D0, Parameter in D1
 70:  aes:
 71:          lea contrl,a0
 72:          move d0,(a0)
 73:          movep.l d1,3(a0)        ;4 Parameter
                                       übertragen
 74:          move.l #aespb,d1
 75:          move #$c8,d0
 76:          trap #2
 77:          rts
 78:
 79:
 80:          data
 81:
 82:  *Pointer auf AES-Arrays
 83:
 84:  aespb:  dc.l contrl,global,intin,intout,addrin,
               addrout
 85:
 86:
 87:          bss
 88:
 89:  *Definitionen der AES-Arrays
 90:
 91:  contrl: ds.w 11
 92:
 93:  global: ds.w 15
 94:
 95:  intin:  ds.w 64
 96:
 97:  intout: ds.w 64
 98:
 99:  addrin: ds.w 64
100:
101:  addrout:ds.w 64
102:
103:  deskinf:ds.b 1024               ;Puffer für
                                       Desktop-Daten
104:
105:  stack:  ds.l 50                 ;sollte für
                                       Stack reichen
```

RESOURCE-EINBINDUNG IN C

Christoph Conrad

Wollen Sie auf dem ATARI ST in Ihrem Programm ein selbsterstelltes Bild, einen neuen Font, eine RSC-Datei oder ein sonstiges Daten-File einbinden? Einige BASIC-Dialekte stellen für solche Zwecke (und für Maschinensprecheroutinen) den INLINE-Befehl zur Verfügung. Was aber z.B. in 'C' und Assembler machen?

Einige Resource-Construction-Sets bieten die Möglichkeit, die Resource-Datei als C-Quelltext auszugeben. Dieses Feature bieten aber nicht alle Resource-Construction-Sets, außerdem erfordert dieser Weg einen mehr oder weniger großen Anpassungsaufwand. Andere Daten-Files können Sie eigentlich nur zur Laufzeit des Programmes dazuladen, oder mit einem Hilfsprogramm in ein C-Quelltext-File umwandeln, in dem ein global initialisiertes char-Array die Bytes des Daten-Files aufnimmt. Eine andere Möglichkeit zeigt das Programm *RSC-2OBJ.TTP* (lies: Resource to Objekt.TTP) auf, die gerade beim Einbinden von Resourcefiles sehr elegant ist. Die vom Programm erwartete Kommandozeile lautet: infile outfile labelname [-r]. *infile* ist der Dateiname des zu konvertierenden Daten-Files. *outfile* ist die konvertierte Objektdatei. *labelname* ist der Name des öffentlichen Labels, unter dessen Adresse Sie Ihre Daten ansprechen (siehe Beispielprogramm).

Um die Arbeitsweise des Konverters zu verstehen, müssen Sie etwas über den Aufbau von Objektdateien, wie Sie Compiler und Assembler erstellen, wissen. Die vorliegenden Betrachtungen sind teilweise etwas vereinfacht und beschränken sich auf den für Sie wesentlichen Aspekt. Das auf dem Atari ST von den gängigen Compilern/Assemblern benutzte Format ist jenes von Digital-Research (GEM stammt auch daher). In diesen Objekt-Files sind die Größen des Codes, der initialisierten und der uninitialisierten Datenbereiche vermerkt sowie die Code- und Datenbereiche abgelegt. Unsere Daten-Files landen komplett im Datensegment der erzeugten Objektdatei; die Größe des Datenbereiches entspricht der des Daten-Files. Codegröße und Größe der uninitialisierten Datenbereiche werden auf Null gesetzt. Weiterhin gibt es eine Symboltabelle, die hauptsächlich Auskunft gibt über die Namen von Variablen, welche Sie aus einem Programm-Modul exportieren bzw. importieren, und die Art der Variablen. Ein Beispiel: Sie definieren in einem C-Modul eine Variable global (außerhalb jeder Funktion), z.B. *int flag*. Diese Variable soll in einem zweiten Programm-Modul benutzt werden und wird dort als *extern int flag* deklariert. Der Linker weiß nun aus der Symboltabelle des Objekt-Files von Modul 2, daß in einem anderen Modul eine Variable *flag* existieren muß und findet diese in der Symboltabelle des Objekt-Files von Modul 1. Der Label-Name, den Sie über die Kommandozeile angegeben haben, wird in der Symboltabelle vermerkt als "global definiert im Datensegment" mit dem Symbolwert 0. Dies bedeutet, daß die Adresse dieses Labels direkt auf die Daten (ohne Versatz) zeigt.

Den für uns wichtigsten Teil des Objekt-Files (zumindest bei RSC-Dateien) stellt die Fixup-Tabelle dar. Programme können zur Laufzeit vom Betriebssystem an ganz unterschiedliche Startadressen geladen werden, eben dort, wo gerade genug Platz frei ist. Aus Codegröße- und Geschwindigkeitsgründen benutzen Compiler (und Assembler-Programmierer) aber viele Maschinenbefehle, die auf eine feste Adresse (z.B. einer Variablen) zugreifen, und zwar so, als begänne der Code/die Daten ab Speicheradresse 0. Das bedeutet z.B.: Falls ein Label relativ zum Codesegmentstart $4711 Bytes entfernt liegt, ist im compilierten Maschinencode (Objektdatei) die Zugriffsadresse $4711. Dem Linker kann aber über die Fixup-Tabelle gesagt werden, daß er für solche Zugriffe Einträge im fertigen Programm in der sogenannten Relokationstabelle macht. Der Lader (das ist die Komponente des Betriebssystems, welches ein Programm von Diskette/Platte lädt und alle nötigen Anpassungen vornimmt) wertet solche Einträge aus und addiert bei allen solchen absoluten Adressen die tatsächliche Startadresse des Codes (er 'reloziert' die Adressen). Die Fixup-Tabelle ist genauso groß wie die Größen von Code- und Datensegment zusammengenommen. Jede zu relozierende Adresse (Länge = 4 Bytes = 1 Long) wird durch 4 Bytes in der Fixup-Tabelle vermerkt, wobei der Abstand des Longs vom Fixup-Tabellenanfang genau dem Abstand des zu relozierenden Longs vom Start des Code- bzw. Datensegments entspricht. Für die Konversion sind zwei beschreibende Werte der Fixup-Tabelle wichtig: 0x00000000 und 0x00050001. Der erste Wert führt dazu, das der Linker in der Relokationstabelle des fertig gelinkten Programms einen Wert einträgt, der dem Lader "bitte dieses Wort (long) nicht verändern!" mitteilt. Der zweite Wert sagt dem Linker, daß es sich um ein zu relozierendes Long im Datensegment handelt, was er entsprechend in der Relokationstabelle berück-

sichtigt. Bei reinen Daten-Files [KEIN Switch -r (!)] wird die gesamte Fixup-Tabelle mit Nullen aufgefüllt, es sollen ja auch keine Daten vom Lader verändert werden. Bei Resourcefiles existieren eine Menge interner Zeiger, z.B. auf TEDINFOs, ICONBLKs, BITBLKs sowie innerhalb dieser Strukturen Zeiger auf Daten etc. (siehe [2],[6]). Alle diese Zeigerwerte beziehen sich aber auf den Anfang der Resource-Datei und werden von *rsrc_load(..)* beim Einladen der Resourcedaten auf die absoluten Speicheradressen durch Addition der RSC-Speicherbasisadresse angepaßt. Na, klingelt's? Das ist ja genau dasselbe wie beim Relozieren! Wenn wir in der Fixup-Tabelle diese Zeiger als relozierbar markieren, wird der Lader uns alle RSC-internen Zeiger setzen.

Die einzige zur Laufzeit notwendige Anpassung besteht in der Umrechnung von Zeichen- in Pixel-Koordinaten. In Resourcefiles sind nämlich die Positionen und Abmessungen der Objekte nicht in Pixeln, sondern in Zeichenbreiten bzw. Zeichenhöhen angegeben, wobei noch ein vorzeichenbehafteter Byteoffset (-128..127) einen Pixel-Versatz angeben kann. Diese Umrechnung kann sehr bequem mittels der AES-Funktion *rsrc_obfix(..)* gemacht werden. Die Funktion *rsrc_gaddr(..)* muß durch eine eigendefinierte Funktion ersetzt werden, da die Originalfunktion auf Daten im global-Array zurückgreift, die durch *rsrc_load* gesetzt werden.

Bei der Konversion von Resourcefiles (Flag -r) wird sicherheitshalber noch ein kleiner Test vorgenommen, ob es sich wirklich um ein Resourcefile handelt, damit beim versehentlichen Setzen dieses Flags normale Daten-Files durch die Relokationsinformationen nicht verunstaltet werden. Aber nicht nur deswegen: Sämtliche relativen Zeiger beziehen sich auf den Resource-Dateistart (wie schon erwähnt) und können deshalb auch nicht negativ sein. Die für die Konversion allokierten Speicherbereiche liegen meist direkt hinter dem Konversionsprogramm im Speicher (an höheren Adressen). Negative Zeiger-Offsets (das sind bei normalen Daten-Files fälschlicherweise als Zeiger-Offsets aufgefaßte Daten) könnten (beim 'Poken' der Fixup-Werte in den dafür allozierten Buffer) das Konversionsprogramm teilweise überschreiben und zu Programmabstürzen führen. Details zum Aufbau der Resource-Dateien würden den Rahmen dieses Artikels bei weitem sprengen. Ich verweise speziell auf [2],[3] und 'die Bibel' [6]. Die Programme wurden in Turbo C 1.0 entwickelt, wobei zunächst eine komplette Implementation in ANSI-Funktionen vorlag. Die I/O-Library von Turbo-C ist jedoch mit einigen herrlichen (nervtötenden) Fehlern versehen, so daß fast zur Gänze auf Gemdos ausgewichen wurde. Als positiver Nebeneffekt ergab sich eine Verkleinerung des fertigen Programmcodes auf ein Drittel der Ursprungsgröße (jetzt ca. 2.5 kByte).

Im Programm setze ich zur Behandlung von Ausnahmesituationen (sprich Fehlern) 'Goto's ein. Da Sie diesen Satz noch lesen, darf ich davon ausgehen, das Sie nicht zu den absoluten Anti-'Goto'-Puristen gehören, welche sicherlich gerade damit beschäftigt sind, diesen Artikel fein säuberlich aus dieser Zeitung herauszutrennen. Daß 'Goto's zum Verlassen tief verschachtelter Kontrollstrukturen gut geeignet sind und eine Menge überflüssiger if-Abfragen ersparen, hat sich ja zum Glück herumgesprochen. Aber auch bei Fehlern ersparen Sie es sich, jedesmal den Ausnahmebehandlungscode durchzulesen, was das Verständnis der eigentlichen Programmabsicht doch erheblich stört. Außerdem sind alle Fehlerbehandlungsmaßnahmen übersichtlich am Ende einer Routine gesammelt.

Zum Beispielprogramm können Sie eine beliebige, mit einem Resource-Construction-Set erstellte und durch den Konverter geschleuste Resource-Datei dazulinken. Der Label-Name sollte *recource* sein. Das Programm zeigt Ihnen alle Objektbäume zentriert auf dem Bildschirm. Menüleisten werden normalerweise nicht zentriert; hier schon, da das Programm nicht wissen kann, ob Ihre Resource eine Menüleiste enthält. Bei Menüs bleibt etwas Pixel-Schrott am oberen Bildschirmrand.



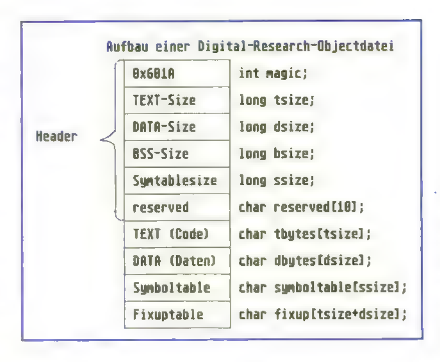


***Bild 1:** Der Aufbau einer Digital-Research-Objektdatei*

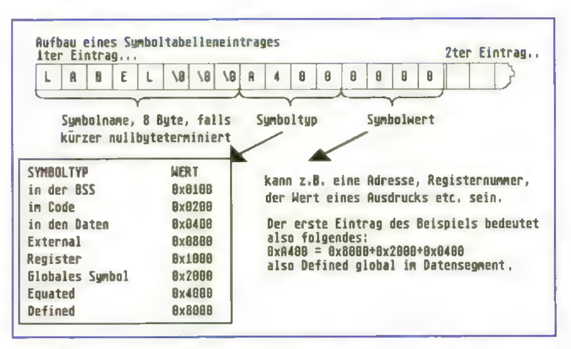


***Bild 2:** Der Aufbau eines Symboltabelleneintrags*

[1] Jürgen Schultz-Kappler, Dialog über Tasten, c't 3/88

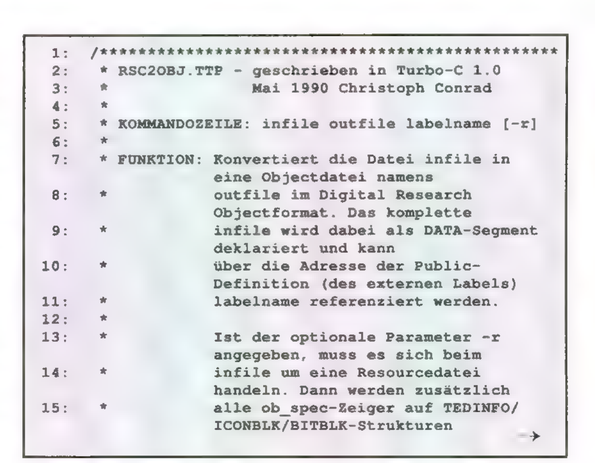
[2] Stefan Höhn, Objektstrukturen im AES, ST-Computer Sonderheft 2 (sehr informativ und verständlich!)

[3] Stefan Höhn, Einführung in das Resource-Construction-Set von Digital Research, ST-Computer Sonderheft No. 2

[4] Jürgen Leonhard, Resource-Datei? Nein danke!, ST-Computer 12/87

[5] Lutz Preßler, Resourcen in GFA-Basic 3.0, ST-Computer 10/88

[6] Jankowski/Reschke/Rabich, ATARI ST Profibuch, Sybex Verlag



```
 1:  /****************************************************
 2:   * RSC2OBJ.TTP - geschrieben in Turbo-C 1.0
 3:   *              Mai 1990 Christoph Conrad
 4:   *
 5:   * KOMMANDOZEILE: infile outfile labelname [-r]
 6:   *
 7:   * FUNKTION: Konvertiert die Datei infile in
                 eine Objectdatei namens
 8:   *          outfile im Digital Research
                 Objectformat. Das komplette
 9:   *          infile wird dabei als DATA-Segment
                 deklariert und kann
10:   *          über die Adresse der Public-
                 Definition (des externen Labels)
11:   *          labelname referenziert werden.
12:   *
13:   *          Ist der optionale Parameter -r
                 angegeben, muss es sich beim
14:   *          infile um eine Resourcedatei
                 handeln. Dann werden zusätzlich
15:   *          alle ob_spec-Zeiger auf TEDINFO/
                 ICONBLK/BITBLK-Strukturen
```

```
16:   *          als zu relozierend vermerkt, die
                 Zeiger innerhalb dieser
17:   *          Strukturen, die Indize der
                 Objectbaumtabelle, der
18:   *          Freestringtabelle, der
                 Freeimagestabelle sowie die bi_pdata-
19:   *          Zeiger der Freeimages.
20:   *          Dies bewirkt, das beim Laden des
                 Programmes durch den
21:   *          GEMDOS-Loader diese Referenzen
                 automatisch auf die absoluten
22:   *          Speicheradressen angepasst werden.
                 Als einzige Arbeit bleibt
23:   *          die Umrechnung von Zeichen- in
                 Pixelkoordinaten mittels
24:   *          rsrc_obfix(...).
25:   *          Der labelname wird immer auf 8
                 Zeichen begrenzt und in
26:   *          Grossbuchstaben konvertiert.
27:   *
28:   * Benötigte Bibliotheken: TOS- und Standardlib.
29:   */
30:
31: #include <tos.h>
32: #include <aes.h>
33:
34: typedef unsigned long size_t;
35:
36: extern void exit(int status);
37: extern char *strupr(char *s);
38: extern int  toupper(int c);
39: extern int  atexit(void (*func)());
40:
41: typedef enum{false, true} bool;
42:
43: /* Vor der Rückkehr des Programmes zum Desktop
44:  * wird auf einen Tastendruck gewartet, damit
45:  * alle Meldungen gelesen werden können.
46:  */
47: void holdScreen(void) {Cconws(„\r\n<Press any
                         key>"); Crawcin();}
48:
49: /* Ausgabe von count Nullbytes auf die Datei
       FILE* file */
50: bool fputNull(int count, int file)
51: {   int byte;
52:
53:     for(byte = 0; byte < count; ++byte)
54:         if(Fwrite(file, 1L, „") != 1L)
55:             return false;
56:     return true;
57: }
58:
59: int main
60:     (int    argc,        /* Anzahl übergebener
                                Parameter + 1 */
61:      char   *argv[])     /* Einzelne Parameter
                          (ab argv[1]) als String */
62: {
63:     int  in, out;              /* Ein/Ausgabedatei -
                                   Filedeskriptoren */
64:     bool rscReloc = false;  /* RSC-
                             Dateikonvertierung ? */
65:     long length;               /* Länge der
                                  Eingabedatei */
66:     char *inbuf;               /* Zeiger auf
                    Einlesebuffer für Eingabedatei */
67:     int  *outbuf;              /* Zeiger auf Buffer
                                 für Fixup-table */
68:   /* Allgemeiner Schleifenindex, long wegen
       Löschen des outbuf (s.u.) */
69:     long i;
70:
71:
72:   /* Bei Programmabschluss soll der
       Ausgabebildschirm bis zu
73:    * einem Tastendruck erhalten bleiben.
74:    */
75:     atexit(holdScreen);
76:
77:   /* Anzahl der Übergabeparameter checken */
78:     if(!(4 <=argc && argc <= 5)) goto Usage;
79:
80:   /* Öffnen der Eingabedatei */
81:     if((in = Fopen(argv[1], 0)) < 0) goto
                                      InfileOpen;
82:
83:   /* Übergabeparameter „-r": Relocation einer
        RSC-Datei an */
84:     if(argv[4] && argv[4][0] == '-' &&
                   toupper(argv[4][1]) == 'R')
85:         rscReloc = true;
86:
87:   /* Dateilänge der Eingabedatei ermitteln */
88:     if((length = Fseek(0L, in, 2)) <= 0 ||
            Fseek(0L, in, 0) != 0)
89:         goto InfileLen;
90:
91:   /* Buffer für Eingabedatei allokieren... */
92:     if(!(inbuf = (char*) Malloc(length))) goto
                              Malloc;
93:
94:   /* ...und Datei einlesen */
95:     if(Fread(in, length, inbuf) != length) goto
                 InfileRead;
96:
97:   /* Test, ob (bei Angabe des Flags -r,
        Resourcedatei konvertieren)
98:    * die im RSC-Dateiheader abgelegte Dateilange
        mit
99:    * der tatsächlichen Länge übereinstimmt.
100:   */
101:    if(rscReloc && ((int*)inbuf)[17] != length)
                            goto NoRSCFile;
102:
103:  /* Öffnen der Ausgabedatei */
104:    if((out = Fcreate(argv[2], 0)) < 0) goto
                                OutfileCreate;
105:
106:  /* Objectdatei-Magic setzen */
107:    if(Fwrite(out, 2, „\x60\x1A") != 2) goto
                                       Write;
108:
109:  /* CODE-Länge == 0 */
110:    if(!fputNull(4, out)) goto Write;
111:
112:  /* DATAlänge = Grösse der Eingabedatei */
113:    if(Fwrite(out, sizeof(long), &length) !=
                  sizeof(long))
114:        goto Write;
115:
116:  /* 4 Nullbytes für BSSlänge
117:   * 3 Nullbytes + 14 (Länge der Symboltabelle
        als Long, ein Eintrag
118:   * in der Symboltabelle = 14 Bytes)
119:   */
120:    if(!fputNull(7, out)) goto Write;
121:    if(Fwrite(out, 1L, „\016") != 1L) goto Write;
122:
123:  /* Zehn reservierte Bytes (Null) */
124:    if(!fputNull(10, out)) goto Write;
125:
126:  /* DATA == Eingabedatei schreiben */
127:    if(Fwrite(out, length, inbuf) != length)
                             goto Write;
128:
129:  /* Der Desktop wandelt ihm übergebene
        Kommandozeilen bei
130:   * TTP-Programmen immer in Grossbuchstaben!!
        strupr hier,
131:   * damit auch bei Kommandozeileninterpretern,
        die dies evt.
132:   * nicht machen, das gleiche Programmverhalten
        auftritt.
133:   */
134:    strupr(argv[3]);
135:
136:  /* Eintrag in der Symboltabelle
137:   * Labelname (8 Bytes) + (0xA400 == Defined
        global in DATA-Segment)
138:   */
139:    for(i = 0; argv[3][i] && i < 8; ++i)
140:        if(Fwrite(out, 1L, &argv[3][i]) != 1L)
              goto Write;
141:    for(; i < 8; ++i)
142:        if(Fwrite(out, 1L, „") != 1L) goto Write;
143:
144:  /* 0xA400 == defined global symbol in DATA-
        Segment */
145:    if(Fwrite(out, 1L, „\xA4") != 1L) goto Write;
146:  /* Ein Nullbyte von 0xA400, vier Nullbytes
        Symbolwert */
147:    if(!fputNull(5, out)) goto Write;
148:
```

→

```
149:      if(!(outbuf = (int*) Malloc(length)))
150:      {   Fdelete(argv[2]);    /* Ausgabedatei
                                    löschen */
151:          goto Malloc;
152:      }
153:      for(i = 0; i < length; ++i) ((char*
154:                    )outbuf)[i] = 0;
155:      if(rscReloc)
156:      /* Resourcedatei, Fixuptabelle aufbereiten */
157:      { /* Zeiger auf Resourcedateiheader */
158:           RSHDR *rsc = (RSHDR*) inbuf;
159:         /* Zeiger auf erstes Object */
160:           OBJECT *obj = (OBJECT*) (rsc->rsh_object
               + (long) rsc);
161:         /* Gesamtzahl der Objecte */
162:           int nrObj = rsc->rsh_nobs;
163:         /* Gesamtzahl der Objektbäume */
164:           int nrObjTrees = rsc->rsh_ntree;
165:         /* Gesamtzahl der Freestrings */
166:           int nrFreeStr = rsc->rsh_nstring;
167:         /* Gesamtzahl der Freeimages */
168:           int nrFreeImg = rsc->rsh_nimages;
169:         /* Abstand eines zu relozierenden Wertes
               vom Resourcedateistart
170:          * in sizeof(int) (sizeof(int), da int *
               outbuf)
171:          */
172:          int dist;
173:
174:         /* Das Fixup-Info sieht HIER immer gleich
               aus: Die Tabelle
175:          * enthält für jedes Long des DATA-
               Segments (== RSC-Datei)
176:          * einen Wert, wobei ein 0L (wegen dem
               calloc) vorgegeben ist
177:          * und „keine Relokation" bedeutet.
178:          * Für alle zu relozierenden Longs wird
               eine 0x00050001 eingetragen,
179:          * wobei das Highword für 'relocatable',
               das Lowword für 'in DATA-
180:          * Segment' steht.
181:          */
182:
183:         /* Objektbaumtabelleneinträge */
184:          dist = rsc->rsh_trindex / 2;
185:          for(i = 0; i < nrObjTrees; ++i)
186:          {   outbuf[dist++] = 5;
187:              outbuf[dist++] = 1;
188:          }
189:
190:         /* Freestringtabelleneinträge */
191:          dist = rsc->rsh_frstr / 2;
192:          for(i = 0; i < nrFreeStr; ++i)
193:          {   outbuf[dist++] = 5;
194:              outbuf[dist++] = 1;
195:          }
196:
197:         /* Freeimagestabelleneinträge + zugehörige
               BITBLK's */
198:          dist = rsc->rsh_frimg / 2;
199:          for(i = 0; i < nrFreeImg; ++i)
200:          {   int bi_pdataDist =
201:                  ( (long*) (rsc->rsh_frimg +
                       (long) rsc) )[i] / 2;
202:
203:            /* bi_pdata-Zeiger in BITBLK's */
204:              outbuf[bi_pdataDist++] = 5;
205:              outbuf[bi_pdataDist] =   1;
206:
207:            /* Tabelleneinträge */
208:              outbuf[dist++] = 5;
209:              outbuf[dist++] = 1;
210:          }
211:
212:         /* Alle Objekte durchgehen */
213:          for(i = 0; i < nrObj; ++i)
214:          { /* Bei ob_type extended-Flags (Bits
                   8..15) ausblenden */
215:              int ob_type = obj[i].ob_type &0xFF;
216:              int j;  /* Lokaler Schleifenindex */
217:
218:            /* Anpassung der ob_spec-Zeiger */
219:              switch(ob_type)
220:              {
221:              case 21: case 22: case 23: case 26:
                                            case 28:
222:              case 29: case 30: case 31: case 32:
223:                  dist = ((long) &obj[i].ob_spec -
                          (long) rsc) / 2;
224:                  outbuf[dist++] = 5;
225:                  outbuf[dist]   = 1;
226:              }
227:
228:            /* Anpassung der Zeiger in TEDINFO/
                 ICONBLK/BITBLK-Strukturen */
229:              switch(ob_type)
230:              {
231:              case 21: case 22: case 29: case 30:
                                            case 31:
232:                /* TEDINFO/ICONBLK */
233:                  dist = (long) obj[i].ob_spec / 2;
234:                  for(j = 0; j < 3; j++)
235:                  {   outbuf[dist++] = 5;
236:                      outbuf[dist++] = 1;
237:                  }
238:                  break;
239:              case 23:
240:                /* BITBLK */
241:                  dist = (long) obj[i].ob_spec / 2;
242:                  outbuf[dist++] = 5;
243:                  outbuf[dist]   = 1;
244:              }
245:          }
246:      }
247:
248:  /* Fixup-Tabelle schreiben */
249:      if(Fwrite(out, length, outbuf) != length)
                                      goto Write;
250:
251:      Fclose(in);
252:      Fclose(out);
253:
254:      Cconws(„Alles paletti!");
255:      return false;
256:
257:      Usage:
258:      Cconws(„Falsche Anzahl Parameter!\r\n");
259:      Cconws(„RSC2OBJ infile outfile labelname [-
                                               r]");
260:      exit(1);
261:
262:      InfileOpen:
263:      Cconws(argv[1]); Cconws(„ konnte nicht
                              geöffnet werden!");
264:      exit(2);
265:
266:      InfileLen:
267:      Cconws(argv[1]); Cconws(„: Fehler beim
                       Ermitteln der Dateilänge!");
268:      exit(3);
269:
270:      Malloc:
271:      Cconws(„Zuwenig Speicher!");
272:      exit(4);
273:
274:      InfileRead:
275:      Cconws(argv[1]); Cconws(„: Fehler beim
                                   Einlesen!");
276:      exit(5);
277:
278:      NoRSCFile:
279:      Cconws(argv[1]); Cconws(„: Keine RSC-Datei
                             oder fehlerhaft!");
280:      exit(6);
281:
282:      OutfileCreate:
283:      Cconws(argv[2]); Cconws(„ konnte nicht
                              geöffnet werden!");
284:      exit(7);
285:
286:      Write:
287:      Cconws(„Fehler beim Schreiben!");
288:      Fdelete(argv[2]);
289:      exit(8);
290:  }
```




```
  1:  /**************************************************
  2:   * RSCTEST.PRG - geschrieben in Turbo-C 1.0
  3:   *             Mai 1990 Christoph Conrad
  4:   *
  5:   * FUNKTION: Test einer RSC-Objectdatei
                Einbindung
  6:   *          durch Anzeigen aller Objektbäume
  7:   *          Der labelname auf der Kommandozeile
                von RSC2OBJ.TTP
  8:   *          muss 'resource' lauten (ohne
                Anführungsstriche)
  9:   *
 10:   * Benötigte Bibliotheken: Standard/Gemlib/TOS-
       Lib.
 11:   */
 12:
 13:  #include <vdi.h>
 14:  #include <aes.h>
 15:  #include <stdio.h>
 16:  #include <stdlib.h>
 17:  #include <tos.h>
 18:
 19:  typedef enum {false, true} bool;
 20:
 21:  /* Die üblichen GEM-Variablen...*/
 22:  int contrl[12], intin[128], intout[128],
          ptsin[128], ptsout[128];
 23:  int work_in[12], work_out[57];
 24:  int handle, phys_handle;
 25:  int gl_hchar, gl_wchar, gl_hbox, gl_wbox;
 26:  int gl_apid;
 27:
 28:  /* Beim AES/VDI anmelden */
 29:  bool open_vwork(void)
 30:  {
 31:      register int i;
 32:
 33:      if((gl_apid = appl_init()) != -1)
 34:      {   for(i = 1; i < 10; work_in[i++] = 0);
 35:          work_in[10] = 2;
 36:          phys_handle = graf_handle(&gl_wchar,
                 &gl_hchar, &gl_wbox, &gl_hbox);
 37:          work_in[0] = handle = phys_handle;
 38:          v_opnvwk(work_in, &handle, work_out);
 39:          return(true);
 40:      }
 41:      else
 42:          return(false);
 43:  }
 44:
 45:  /* Beim VDI/AES abmelden */
 46:  void close_vwork(void)
 47:  {   v_clsvwk(handle);
 48:      appl_exit();
 49:  }
 50:
 51:  extern RESOURCE;
 52:  /* rsc zeigt auf den Anfang der Resourcedatei,
        also den Resourceheader */
 53:  RSHDR *rsc = (RSHDR*) &RESOURCE;
 54:
 55:  /* Ersatz der Funktion rsrc_gaddr des AES.
 56:   * Es werden  Objektbäume/Freestrings/Freeimages
        berücksichtigt.
 57:   * Die Original-AES-Funktion liefert bei anderen
        Typen zum Teil
 58:   * seltsame Ergebnisse.
 59:   * Die Positionsoffsets im Resourceheader
        verweisen auf (relozierte)
 60:   * Adresstabellen mit Zeigern auf die
        eigentlichen Objekte.
 61:   * rsrc_gaddr liefert bei
 62:   *  - Objectbäumen (re_gtype == 0): Die
          Anfangsadresse des Baums
 63:   *  - Free-Strings/Images: Zeiger auf Zeiger auf
          das eigentliche Object
 64:   */
 65:  int rsrc_gaddr(int re_gtype, int re_gindex,
                     OBJECT **re_gaddr)
 66:  {   if(re_gtype == 0 && 0 <= re_gindex &&
          re_gindex < rsc->rsh_ntree)
 67:      {   *re_gaddr = *((OBJECT**)
 68:                      (rsc->rsh_trindex + (long)
                          rsc + 4 * re_gindex));
 69:          return true;
 70:      }
 71:
 72:      if(re_gtype == 15 && 0 <= re_gindex &&
          re_gindex < rsc->rsh_nstring)
 73:      {   *re_gaddr = (OBJECT*)    /* eigentlich
                                        OBJECT** */
 74:                  (rsc->rsh_frstr + (long) rsc
                          + 4 * re_gindex);
 75:          return true;
 76:      }
 77:
 78:      if(re_gtype == 16 && 0 <= re_gindex &&
          re_gindex < rsc->rsh_nimages)
 79:      {   *re_gaddr = (OBJECT*)    /* eigentlich
                                        OBJECT** */
 80:                  (rsc->rsh_frimg + (long) rsc
                          + 4 * re_gindex);
 81:          return true;
 82:      }
 83:
 84:      return false;
 85:  }
 86:
 87:  /* Für alle Objecte wird eine Transformation von
 88:   * Zeichen in Pixelkoordinaten vorgenommen.
 89:   */
 90:  void rsrc_AllFix(OBJECT *firstObj, int nrObj)
 91:  {   int obj;
 92:
 93:      for(obj = 0; obj < nrObj; ++obj)
 94:          rsrc_obfix(firstObj, obj);
 95:  }
 96:
 97:  bool main(void)
 98:  {
 99:      if(open_vwork() == true)
100:      {   OBJECT *objTree;
101:          int nrObj = rsc->rsh_nobs;
102:          int x,y,w,h;
103:          int nrObjTrees = rsc->rsh_ntree;
104:          int actObjTree;
105:
106:          graf_mouse(0,0);           /* Maus als
                                          Pfeil */
107:
108:          rsrc_gaddr(0,0,&objTree);   /* Adresse
                     des ersten Objectbaums */
109:          rsrc_AllFix(objTree, nrObj);/* Zeichen -
                  > Pixelkoordinaten */
110:
111:          for(actObjTree = 0; actObjTree <
              nrObjTrees; actObjTree++)
112:          {
113:            /* Objektbaumadresse bestimmen */
114:              rsrc_gaddr(0, actObjTree, &objTree);
115:            /* Zentrieren des Objektbaums
116:             * Menüleisten sind nicht zum
                 Zentrieren gedacht, bei
117:             * ihnen bleibt Müll am oberen
                 Bildschirmrand nach
118:             * dem Restaurieren mit form_dial(3,
                 ..).
119:             */
120:             form_center(objTree, &x, &y, &w, &h);
121:
122:            /* Bildschirmbereich reservieren */
123:              form_dial(0, x, y, w, h, x, y, w, h);
124:            /* Objektbaum zeichnen */
125:             objc_draw(objTree, 0, 8, x, y, w, h);
126:                  Crawcin();
127:            /* Bildschirmbereich freigeben */
128:             form_dial(3, x, y, w, h, x, y, w, h);
129:          }
130:
131:          close_vwork();
132:          return false;
133:      }
134:      else
135:      {   printf(„Fehler bei der
                  Programminitialisierung!");
136:          return true;
137:      }
138:  }
```

FARBPALETTEN-WECHSEL

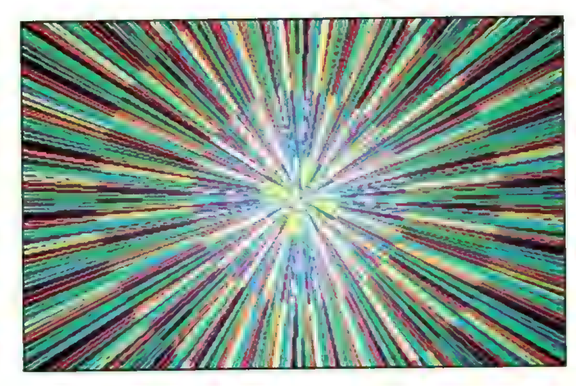
Christen Fihl

DAS PROGRAMM, DAS IN DIESEM ARTIKEL VORGESTELLT WIRD, SOLL ZEIGEN, WIE SIE IN IHREN PROGRAMMEN FARBPALETTEN-ÄNDERUNGEN DAZU BENUTZEN KÖNNEN, UM Z.B. ZEICHNUNGEN ZU ANIMIEREN ODER SCHÖNE FARBEFFEKTE AUF DEM MONITOR DARZUSTELLEN.

Wenn man auf dem Bildschirm zeichnet, ist es notwendig, die Farbe anzugeben, mit der gezeichnet werden soll. Es ist die Farbnummer (0-15), die, wenn sie auf dem Bildschirm angezeigt wird, durch die Benutzung der Palette umgewandelt wird. Weil diese Aktion mit sehr hoher Geschwindigkeit abläuft, sind Veränderungen in der Palette sofort auf dem Bildschirm zu erkennen.

Für Animationen ist es charakteristisch, daß Bilder auf dem Monitor schnell wechseln, ohne daß man sieht, wie sie aufgebaut werden. Dieser Effekt ist z.B. dadurch möglich, daß man die Palette ändert. Natürlich dauert es immer noch genau so lange, ein Bild aufzubauen, aber der Effekt läßt sich für den Anwender des Programms unsichtbar programmieren.

Um unsichtbare Linien zu ziehen, muß man eine Farbe verwenden, die man vom Hintergrund nicht unterscheiden kann. Deshalb ist es notwendig, die Anzahl der Farben auf zwei oder vier zu beschränken. Im Beispiel-Animationsprogramm werden vier Farben benutzt, was bedeutet, daß zwei Figuren animiert werden können (4*4 = 16 Farben). Durch die Benutzung von lediglich 2 Farben lassen sich vier Figuren animieren (2*2*2*2).



Nehmen wir an, daß zwei Figuren auf dem Bildschirm zur Darstellung kommen. Wir wollen die Bilder jeweils einzeln anzeigen. Das erste Bild wird in den Farben 0, 1, 2 und 3 dargestellt, das andere in den Farben 0, 4, 8 und 12. Beim Aufbau des Bilds muß der Schreibmodus entweder auf OR oder auf XOR gestellt werden. Dadurch verhindert man, daß das zweite Bild das erste zerstört.

Durch das Tauschen der Paletten (0,9,2,3,0,9,2,3,0,9,2,3,0,9,2,3), (0,0,0,0,9,9,9,9,2,2,2,2,3,3,3,3) und (0,9,2,3,2,9,2,3,2,9,2,3,0,9,2,3) können die beiden Bilder separat angezeigt werden. Diese Werte kamen zustande durch das Prüfen der Bitpatterns, die mit 4 Bits möglich sind. Das erste Bild benutzt die Bits XX, während das andere die Bits YY verwendet. Die möglichen Kombinationen sind dabei XX00, XX01, XX11, XX10, 00YY, 01YY, 10YY und 11YY.

Die *Graph unit*-Routine *GetPalette(var Pal:PaletteType)* meiner Maschine gibt die folgenden Werte zurück: (0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15). 0 ist dabei schwarz, 1 = ...

Das Animationsprogramm schaltet zwischen zwei Bildern hin und her. Das Palettenprogramm erstellt einige psychedelische Grafiken. Beide Programme setzen natürlich einen Farbmonitor voraus.

```
 1:  Program PaletteDemo; (* MAXON Pascal *)
 2:
 3:  Uses Graph;
 4:
 5:  Const
 6:          MulFact = 0.0174532925;
 7:  Var
 8:          o,p,m,a,
 9:          Hx,Hy,
10:          Driver,
11:          Mode     : Integer;
12:          Rad      : Extended;
13:          Pal      : PaletteType;
14:  Begin
15:    Driver := StColor;
16:    Mode   := StLow;
17:    InitGraph(Driver,Mode,'');
18:    Hx := GetMaxX DIV 2;
19:    Hy := GetMaxY DIV 2;
20:
21:    { Draw lines }
22:    For A := 1 to 360 Do
23:      Begin
24:        Rad := MulFact*A;
25:        MoveTo(Hx,Hy);
26:        O := A Mod GetMaxColor;
27:        For P := 1 to 4 Do
28:          Begin
29:            SetColor(O);
30:            M := 8+15*P;
31:            LineRel(Round(M*Sin(Rad)),Round(M*
                   Cos(Rad)));
32:            O := GetMaxColor-O;
33:          End;
34:      End;
35:
```

```
36:     { Play with palette }
37:     GetPalette(Pal);
38:     With Pal Do
39:      Repeat
40:       Delay(50);
41:       M := Colors[1];
42:       For A := 1 to GetMaxColor-1 do
43:         Colors[A] := Colors[A+1];
44:       Colors[GetMaxColor] := M;
45:       SetAllPalette(Pal);
46:     Until keypressed;
47:     CloseGraph;
48:   End.
```

```
 1:   Program Animate;   (* MAXON Pascal *)
 2:
 3:   uses Graph;
 4:
 5:   Const
 6:     Snow=false;
 7:   Var
 8:     n,driver,mode  : Integer;
 9:     P1,P2,P3       : PaletteType;
10:
11:   begin
12:     FillChar(P1,SizeOf(P1)*3,0);
13:     With P1 Do begin
14:       Size := 16;
15:       Colors[1]:=9;Colors[2]:=2;Colors[3]:=3;
          Colors[5]:=9;Colors[6]:=2;
16:       Colors[7]:=3;Colors[9]:=9;Colors[10]:=2;
          Colors[11]:=3;Colors[13]:=9;
17:       Colors[14]:=2;Colors[15]:=3;
18:     End;
19:     With P2 Do begin
20:       Size := 16;
21:       Colors[4]:=9;Colors[5]:=9;Colors[6]:=9;
          Colors[7]:=9;Colors[8]:=2;
22:       Colors[9]:=2;Colors[10]:=2;Colors[11]:=2;
          Colors[12]:=3;Colors[13]:=3;
23:       Colors[14]:=3;Colors[15]:=3;
24:     End;
25:     With P3 Do begin
26:       Size := 16;
27:       Colors[1]:=9;Colors[2]:=2;Colors[3]:=3;
          Colors[4]:=2;Colors[5]:=9;
28:       Colors[6]:=2;Colors[7]:=3;Colors[8]:=2;
          Colors[9]:=9;Colors[10]:=2;
29:       Colors[11]:=3;Colors[13]:=9;Colors[14]:=2;
          Colors[15]:=3;
30:     End;
31:
32:     Driver := StColor;
33:     Mode   := StMedium;
34:     InitGraph(Driver,Mode,'');
35:
36:     SetWriteMode(XORPut);
37:     SetColor(2); Rectangle(220,10,250,40);
                     {Control boxes}
38:     SetColor(8); Rectangle(220+50,10,250+50,40);
39:     repeat
40:       n:=1-1; SetColor(8); Rectangle(n+1,n+1,n+181,
                                          n+181);
41:       for n:=1 to 20 do begin
42:         SetColor(2); Rectangle(n,n,n+180,n+180);
43:         if not Snow then SetAllPalette(P3);
                                {Show both}
44:         SetAllPalette(P1); Delay(10);
45:         SetColor(8); Rectangle(n,n,n+180,n+180);
46:         SetColor(8); Rectangle(n+1,n+1,n+181,n+
                                     181);
47:         if not Snow then SetAllPalette(P3);
              {Prevent „holes"}
48:         SetAllPalette(P2); Delay(10);
49:         SetColor(2); Rectangle(n,n,n+180,n+180);
50:       end;
51:       n:=20+1; SetColor(8); Rectangle(n,n,n+180,n+
                                            180);
52:     until keypressed;
53:     closegraph;
54:   end.
```

MOVEDIAL

Bewegliche Dialogboxen



Da steht er nun, der neue Großbildschirm für „meinen" ATARI; schön sieht er aus, eine echte Zierde für den Schreibtisch. Der Kaffee wird gerade fertig, und ich genehmige mir eine Tasse. Jetzt aber schnell das Getränk beiseite stellen und die Festplatte nach geeigneten Programmen zum Testen der Großmonitorfähigkeiten durchforsten. Toll, wie gut mit Programmen wie Calamus oder Arabesque zu arbeiten ist, wenn erstmal die Arbeitsfläche groß genug ist. Blöde ist nur, daß ich bei jedem Öffnen einer Dialogbox mit der Maus Richtung Bildschirmmitte fahren muß, um diese bedienen zu... Schei.., das war die Kaffeetasse; der schöne Computer, ersoffen in Kaffee; und wie das stinkt! Mit dem Standardmonitor wäre das nicht passiert; blöder Großmonitor.

Was können wir dieser aus dem Leben gegriffenen Geschichte (leider nicht aus meinem Leben, da noch kein Besitzer eines Großmonitors) an Weisheit entnehmen: Nun, Kaffee ist nicht nur für Menschen ungesund. Nein, im Ernst, es wäre doch eine tolle Sache, wenn man sich seine Dialogboxen so auf dem Bildschirm anordnen könnte, daß man optimal damit arbeiten kann. Keine Hin- und Herfahrerei mit der Maus mehr, es lebe der kleine Schreibtisch! Daneben gibt es aber noch andere, ungeahnte Möglichkeiten. Denn oft wird auch ein gerade wichtiger Desktop-Teil von einer unvermittelt auftauchenden Dialogbox überdeckt, die auch noch unverschämterweise nach gerade den verdeckten Informationen fragt. Dann kann man wieder auf *ABBRUCH* drücken (wenn's geht), nachsehen, was darunter steht, dieses auswendig lernen oder aufschreiben und daraufhin die ganze Prozedur von neuem starten. Tusch...., dem muß aber nicht so sein; es gibt ja *MOVEDIAL*, die verschiebbaren Dialogboxen. Mit der *MOVEDIAL-Library* können Dialogboxen erstellt werden, die frei auf dem Bildschirm verschiebbar sind oder wahlweise an der aktuellen Mausposition erscheinen (auf neuhochdeutsch Pop-Up-Menüs genannt). Für C-Programmierer (nicht nur) werden einfach zu handhabende Dialogbox-Routinen bereitgestellt, die komplett den Aufruf, die Darstellung und Verwaltung folgender drei Dialogboxtypen übernehmen:

- **Standard-Dialogboxen:** diese erscheinen wie gewohnt in der Bildschirmmitte und sind nicht verschiebbar.

- **Movedial-Boxen:** erscheinen beim ersten Aufruf ebenfalls in der Bildschirmmitte, können dann aber vom Anwender beliebig auf dem Desktop verschoben (gedragged) werden. Bei einem erneuten Aufruf erscheinen sie dann wieder an der letzten Position.

- **Pop-Up-Dialogboxen:** diese tauchen immer dort auf, wo sich gerade der Mauszeiger befindet.

Die *MOVEDIAL-Library* kann von interessierten und fachkundigen Anwendern noch erweitert werden, worauf am Ende des Artikels noch spezieller eingegangen wird.

Theorie

Nach diesem lockeren Einstieg soll ein wenig Theorie (ganz wenig und ganz einfach - Pfadfinderehrenwort) der Objektprogrammierung die Idee der *MOVEDIAL-Library* verständlicher machen. Wer allerdings mit der Programmierung von Objekten (AES-Objekten des GEMs) auf Du und Du steht, kann diesen Teil beruhigt überspringen, denn hier erfährt er nichts Neues (aber später wieder weiterlesen! Versprochen?).

Objekte sind die Grundelemente des ganzen AES-Systems. Die Drop-Down-Menüs, Icons, Fenster, Alert-Boxen und auch die hier behandelten Dialogboxen bestehen aus nichts anderem als Objekten (na ja, ein wenig Hirnschmalz ist auch noch drin; nicht immer, aber manchmal). *Leser:* Jetzt weiß ich aber immer noch nicht was Objekte sind. *Autor:* Nun, ganz einfach: Objekte sind nichts anderes als Datenstrukturen, die alle grafischen Objekte beschreiben. In Abbildung 1 ist diese Datenstruktur in C-Notation zu erkennen. Alle Objekte sind in einem Objektbaum zusammengefaßt. In diesem Baum gibt es Nachbarn (*ob_next*) und Kinder (*ob_head* und *ob_tail*). Kinder können wieder Kinder haben, diese wieder Kinder usw. Nachbarn in dem Objektbaum sind z.B. die verschiedenen Dialogboxen eines Programms. Kinder innerhalb einer Dialogbox stellen die Texte, Eingabefelder und Buttons dar, die selbst wieder Objekte sind. Damit man sich das alles ein wenig besser vorstellen kann, soll es an einem Beispiel verdeutlicht werden. In Abbildung 2 ist eine Dialogbox zu sehen, in der zwei Kästen mit Buttons enthalten sind (abgesehen vom *OK*-Button). Die beiden Kästen stellen Nachbarn (innerhalb der Dialogbox) dar, mit den Auswahlknöpfen als Kinder. Das Schöne an der Sache ist nun (deshalb erzähle ich das ganze hier überhaupt), daß durch Angabe der Position (*ob_x* und *ob_y*) eines der Nachbar-Objekte (siehe Abbildung 1) innerhalb des Vater-Objekts (gemeint ist das übergeordnete Objekt, hier der Dialogbox-Rahmen) verschoben werden kann und gleichzeitig alle Kinder-Objekte (hier die Buttons) mit verschoben werden. Das liegt daran, daß alle Positionsangaben relative Positionen zum Vater-Objekt darstellen. Bei Verschiebung des Vater-Objekts wird logischerweise das Kind-Objekt mit verschoben.

```
typedef struct
{
        int             ob_next;   /* -> nächste Objekt (Nachbar) */
        int             ob_head;   /* -> erstes Kind              */
        int             ob_tail;   /* -> letztes Kind             */
        unsigned int    ob_type;   /* Objektart                   */
        unsigned int    ob_flags;  /* diverse Flags               */
        unsigned int    ob_state;  /* Objektstatus                */
        char            *ob_spec;  /* Zeiger auf weitere Struktur */
        int             ob_x;      /* x-Position des Objekts      */
        int             ob_y;      /* y-Position des Objekts      */
        int             ob_width;  /* Breite des Objekts          */
        int             ob_height; /* Hohe des Objekts            */
} OBJECT;
```

Abb. 1

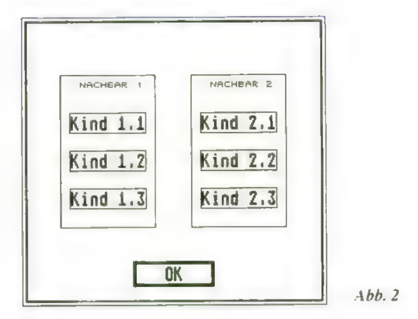


Abb. 2

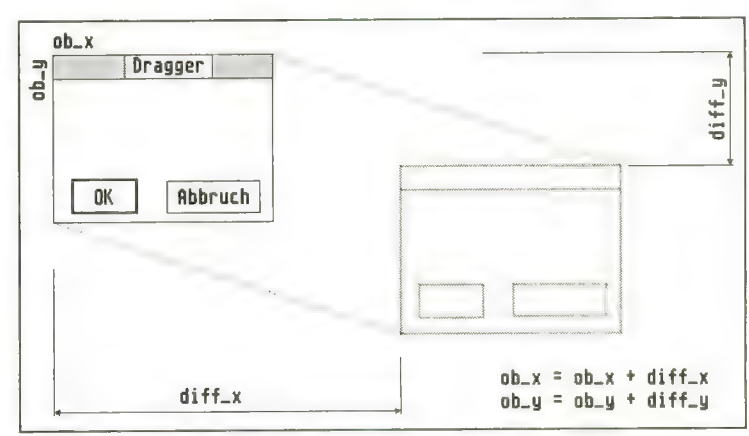


Abb. 3

(Jetzt sollten auch die *Programmierprofis* wieder weiterlesen). Genau dieser Umstand (die relativen Positionsangaben) läßt sich nun zum Verschieben der Dialogbox nutzen. Normalerweise wird mit einem *form_center()*-Aufruf vor dem Zeichnen des Dialogs eine Zentrierung der Dialogbox auf dem Desktop vorgenommen. Dieser *form_center()*-Aufruf berechnet aus der Desktop- und der Dialogboxgröße ein Koordinatenpaar (*ob_x* und *ob_y*) und trägt dies in die Objektstruktur (Abbildung 1) ein. Unterläßt man das Zentrieren, wird der Objektbaum (also die Dialogbox) an der in der Objektstruktur angegebenen Position (*ob_x* und *ob_y*) relativ zur Desktop-Position ausgegeben. Legt man in jeder Dialogbox einen speziellen *Drag*-Button an (ähnlich der Move-Leiste bei Fenstern, die dem Verschieben dient), kann man nach Druck auf diesen Button (linke Maustaste festhalten!) den Dialog verschieben. Dazu müssen Sie den Drag-Button als *EXIT*-Button deklarieren, worauf nach dessen Anwahl der Dialog verlassen wird. Jetzt kann die Wegdifferenz, die mit gedrücktem Mausknopf zurückgelegt wurde (die Wegdifferenz der Maus wohlgemerkt), erfaßt und zur alten Objektposition addiert werden. Dadurch setzen Sie die Dialogbox auf die neue gewünschte Position. Da sich dieser Vorgang recht kompliziert anhört, soll er durch Abbildung 3 bildhaft dargestellt werden (ein Bild sagt mehr als tausend Worte!). Nach diesen Vorbemerkungen sollen nun

die Besonderheiten der einzelnen Dialogboxarten der *MOVEDIAL-Library* erläutert werden. Wie der Leser sicher schon geargwöhnt hat, bauen die einzelnen Dialogboxroutinen auf einem gemeinsamen Unterprogrammstamm auf. Diese Routinen dienen zum Vorbereiten und Zeichnen der Dialogbox sowie zum Restaurieren des Bildschirms und benützen ausschließlich GEM-Routinen. Das ist der einzig richtige Weg, um auflösungsunabhängige Programme zu entwickeln, die dann auf jeder vom GEM unterstützten Grafikkarte lauffähig sind. Die genannten Routinen sind im einzelnen weiter unten beschrieben. Als erstes soll der *Standard-Dialog* erwähnt werden. Eigentlich gibt es hierzu nichts mehr Besonderes zu sagen, da weiter oben schon alles Wichtige über die hier verwandte *form_center()*-Prozedur geschrieben wurde. Durch diese Routine wird die Dialogbox auf dem Desktop mittig zentriert. Über die *Movedial-Boxen* gibt es dagegen einiges mehr zu sagen. Nachdem nach obengenanntem Verfahren (Abbildung 3) der Verschiebungsvektor (Differenzwerte in x- und y-Richtung) ermittelt ist, kann die Dialogbox an der alten Position gelöscht und an der neuen wieder ausgegeben werden. Warum aber müssen die Differenzwerte ermittelt werden und nicht eine absolute Position? Nun, als Grundlage für die neue Position der Dialogbox wird die aktuelle Mausposition nach Loslassen der linken Maustaste genommen. Würde die Dialogbox nun an diese Position gesetzt werden, wäre die linke obere Ecke genau unter dem Mauszeiger. Das war aber nicht beabsichtigt. Man hatte die Dialogbox ja irgendwo innerhalb des Drag-Buttons geschnappt und verschoben; dann soll sie gefälligst auch wieder so gezeichnet werden, daß der *Drag*-Button genauso unter der Maus zum Liegen kommt wie vor der Verschiebung. Genau das erreicht man durch Addition des Verschiebungsvektors auf die alte Dialogboxposition (siehe nochmals Abbildung 3). Eine Bereichsüberschreitung, also ein Verlassen des Desktop-Bereichs beim Verschieben der Dialogbox, ist nicht möglich, da die den Verschieberahmen zeichnende AES-Betriebssystemroutine *graf_dragbox()* die Maximalkoordinaten mit übergeben bekommt. Die Position einer *Pop-Up-Dialogbox* wird aus der aktuellen Mausposition berechnet, d.h. eine solche Box erscheint immer zentriert um die momentane Mausposition. Dabei gilt es natürlich noch die Fälle einer hier möglichen Bereichsüberschreitung abzufangen, denn die Dialogbox soll ja immer innerhalb des Desktops auftauchen, damit alle Elemente auch mit der Maus erreichbar sind.

```
 1: /***************************************/
 2: /*          L I S T I N G  1           */
 3: /***************************************/
 4: /*  MOVEDIAL-Library für Dialogboxen  */
 5: /*  written in Turbo C 2.0             */
 6: /*-------------------------------------*/
 7: /*  Sourcefile: MOVELIB.C              */
 8: /*-------------------------------------*/
 9: /*  by M.Baldauf 6/1990                */
10: /*  (c) MAXON Computer                 */
11: /***************************************/
12:
13: #include <aes.h>          /* AES-Routinen */
14: #include <vdi.h>          /* VDI-Routinen */
15: #include "movelib.h"      /* Unterprogramm-
     Deklarationen */
16:
17: /***********************************************/
18:
19: /*+-----------------------------------------+*/
20: /*| Erledigung der leidigen GEM-Initialisierung
     zu Anfang            |*/
21: /*+-----------------------------------------+*/
22: int gem_init(void)
23: {
24:   int handle;
25:   int ap_id,work_in[12],work_out[57];
26:   int gr_1,gr_2,gr_3,gr_4, i;
27:
28:   /* Applikation anmelden */
29:   ap_id = appl_init();
30:
31:   /* war alles OK? */
32:   if (ap_id==-1)
33:     /* scheint nicht so */
34:     return(-2);
35:   else
36:   {
37:     /* scheinbar war alles OK! => VDI-Handle
        ermitteln */
38:      handle = graf_handle(&gr_1,&gr_2,&gr_3,&gr_4);
39:
40:     for (i=0; i<10; work_in[i++]=1)
41:       ;
42:     work_in[10] = 2;
43:
44:     /* Virtual-Workstation öffnen */
45:     v_opnvwk(work_in, &handle, work_out);
46:
47:     /* und das ermittelte Handle zurückliefern */
48:     return(handle);
49:   }
50: }
51:
52: /*+-----------------------------------------+*/
53: /*| Abmelden der Applikation vom GEM
                        |*/
54: /*+-----------------------------------------+*/
55: void gem_exit(int handle)
56: {
57:   /* Virtual-Workstation schliePen */
58:   v_clsvwk(handle);
59:
60:   /* Applikation abmelden */
61:   appl_exit();
62: }
63:
64: /***********************************************/
65:
66: /*+-----------------------------------------+*/
67: /*| Vorbereitungen zur Dialogbox-Darstellung  |*/
68: /*+-----------------------------------------+*/
69: void vor_dial(OBJECT *dialbox, int dialtype)
70: {
71:   GRECT a;
72:   int mx,my,dummy;
73:   GRECT desk;
74:
75:   /* Fenstererneuerung */
76:   wind_update(BEG_UPDATE);
77:
78:   switch (dialtype)
79:   {
80:     case 1: /* MOVEDIAL */
81:             /* war dies der erste Aufruf? */
82:             if ((dialbox->ob_x == 0)
                   && (dialbox->ob_y == 0))
```

→



MOVEDIAL-Library

Nach den theoretischen Grundlagen sollen die einzelnen realisierten Routinen der *MOVEDIAL-Library* (Listing 1 und Listing 2) erläutert werden. Da die Programme in Turbo C (mit Version 1.1 und 2.0 getestet) entwickelt wurden, war es notwendig, die einzelnen Unterprogramme zu deklarieren (Prototyping). Diese Prototypen wurden in das Include-File *MOVEDIAL.H* (Listing 2) verfrachtet, das beim Compilieren der Library mit eingelesen wird. Nach den üblichen GEM-An- und Abmelderoutinen [*gem_init()* und *gem_exit()* in Listing 1] können die eigentlichen Dialogbox-Routinen besprochen werden. In der Prozedur *vor_dial()* werden die Vorbereitungen zur Darstellung der Dialogbox getroffen. Übergeben wird ein Zeiger auf den Dialogbaum und der Dialogtyp. Es werden dabei folgenden Typen (*dialtype*) unterschieden:

1 - Movedial-Boxen
2 - Pop-Up-Dialogbox
andere - Standard-Dialogboxen

In dieser Prozedur [*vor_dial()*] werden nach den oben besprochenen Methoden die Zeichenkoordinaten der Dialogbox (*ob_x* und *ob_y*) relativ zum Desktop ermittelt. Danach kann mit dem *form_dial()*-Aufruf der notwendige Bildschirmhintergrund reserviert werden. Mit Hilfe der Unterroutine *draw_dial()* wird dann der Dialogbaum auf dem Bildschirm ausgegeben. Der Aufruf *objc_draw()* zeichnet den Objektbaum nur, verwaltet ihn aber nicht. Die Verwaltung wird in den weiter unten beschriebenen Routinen *do_dial()*, *do_movedial()* und *do_Pop-Up()* vorgenommen. Nachdem eine Dialogbox wieder verlassen wurde, muß der Bildschirmhintergrund auch wieder freigegeben werden. Das erfolgt durch die Prozedur *nach_dial()*. Als nächstes kann die interessanteste Routine *movedial()* unter die

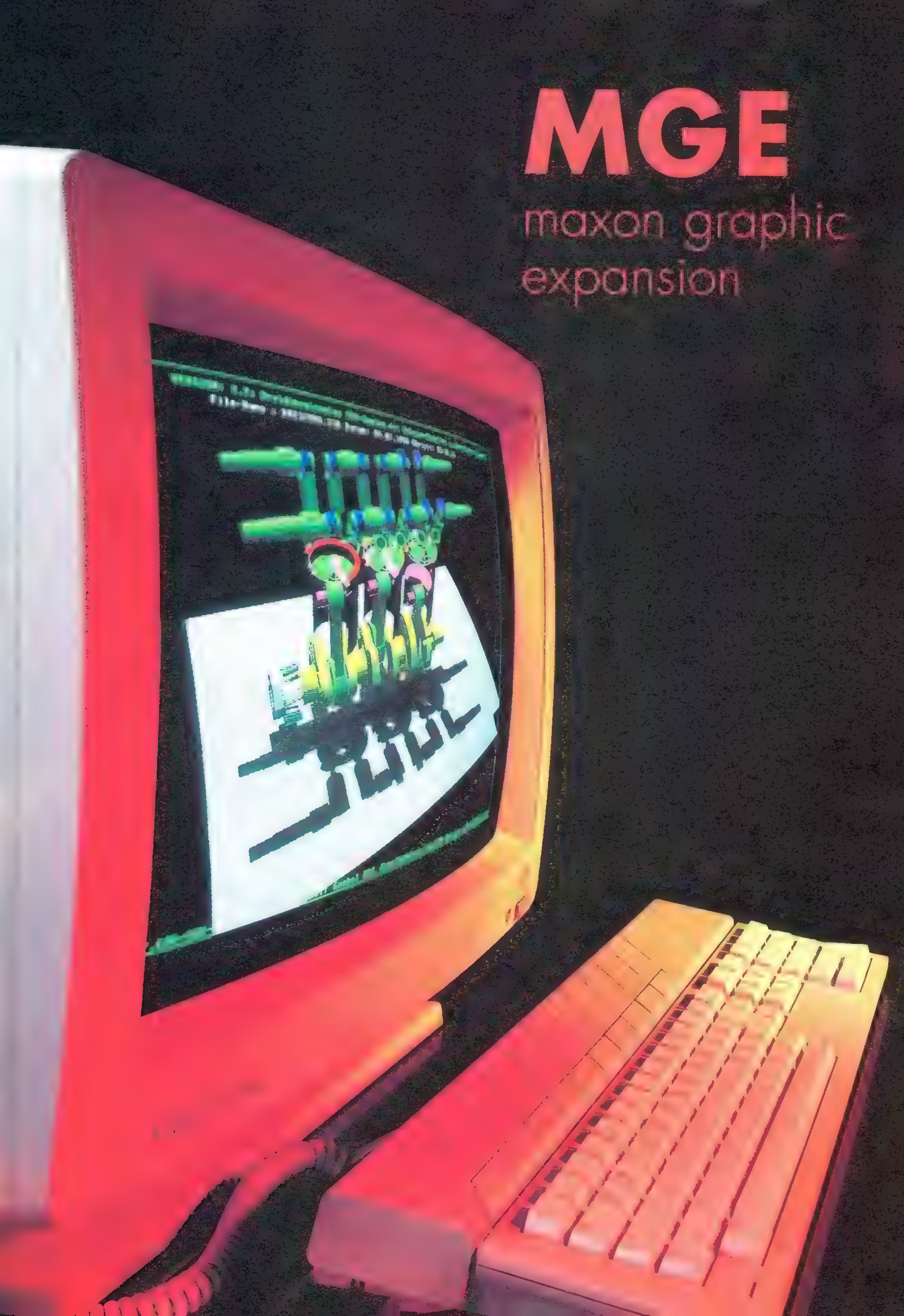
```
 83:            {
 84:              form_center(dialbox,&a.g_x,&a.g_y,
                              &a.g_w,&a.g_h);
 85:              dialbox->ob_x = a.g_x;
                  dialbox->ob_y = a.g_y;
 86:              dialbox->ob_width = a.g_w;
                  dialbox->ob_height = a.g_h;
 87:            }
 88:            else
 89:            {
 90:              a.g_x = dialbox->ob_x;
                  a.g_y = dialbox->ob_y;
 91:              a.g_w = dialbox->ob_width;
                  a.g_h = dialbox->ob_height;
 92:            }
 93:            break;
 94:
 95:    case 2: /* POPUPDIAL */
 96:            /* Mausposition ermitteln */
 97:            dummy = evnt_button(1,1,0,&mx,&my,
                                    &dummy,&dummy);
 98:
 99:            /* GröPe des Desktop ermitteln */
100:            wind_get(0,WF_WORKXYWH,&desk.g_x,
                 &desk.g_y,&desk.g_w,&desk.g_h);
101:
102:            /* X-Position der Dialogbox
                   bestimmen */
103:            a.g_x = mx - (dialbox->ob_width / 2);
104:            /* ist sie innerhalb der Desktop-
                   Grenzen? */
105:            if (a.g_x < desk.g_x)
106:              a.g_x = desk.g_x;
107:            if ((a.g_x + dialbox->ob_width)
                  >(desk.g_x + desk.g_w))
108:              a.g_x = desk.g_w - dialbox
                          ->ob_width;
109:            /* Objekt-Position neu setzen */
110:            dialbox->ob_x = a.g_x;
111:
112:            /* Y-Position der Dialogbox
                   bestimmen */
113:            a.g_y = my - (dialbox
                          ->ob_height / 2);
114:            /* ist sie innerhalb der Desktop-
                   Grenzen? */
115:            if (a.g_y < desk.g_y)
116:              a.g_y = desk.g_y;
117:            if ((a.g_y + dialbox->ob_height)
                  >(desk.g_y + desk.g_h))
118:              a.g_y = desk.g_h - dialbox-
                  >ob_height;
119:            /* Objekt-Position neu setzen */
120:            dialbox->ob_y = a.g_y;
121:
122:            break;
123:
124:    default:/* STANDARDDIAL */
125:            /* Dialogbox zentrieren */
126:            form_center(dialbox,&a.g_x,&a.g_y,
                            &a.g_w,&a.g_h);
127:            break;
128:  }
129:
130:  /* Hintergrund reservieren */
131:  form_dial(FMD_START, a.g_x,a.g_y,a.g_w,a.g_h,
                a.g_x,a.g_y,a.g_w,a.g_h);
132: }
133:
134: /*+-------------------- --------------------+*/
135: /*| Dialogbox zeichnen (nur nach 'vor_dial'
       aufrufen !!!)                           |*/
136: /*+----------------------------------------+*/
137: void draw_dial(OBJECT *dialbox, int dialtype)
138: {
139:   GRECT a;
140:
141:   switch (dialtype)
142:   {
143:     case 1: /* MOVEDIAL */
144:             a.g_x = dialbox->ob_x;
                a.g_y = dialbox->ob_y;
145:             a.g_w = dialbox->ob_width;
                a.g_h = dialbox->ob_height;
146:             break;
147:     case 2: /* POPUPDIAL */
148:             a.g_x = dialbox->ob_x;
```

→

Lupe genommen werden. Hier wird die Dialogbox (nur, wenn es sich um eine MOVEDIAL-Box handelt) von der alten auf eine vom Benutzer gewünschte neue Position verschoben. Nachdem die Maus die Kontrolle übernommen hat [*wind_update()*] und die Form einer Patschehand (*FLAT_HAND*) bekommen hat, wird die Größe des Desktops ermittelt [mittels *wind_get()*]. Jetzt kann die Dialogbox innerhalb der Desktop-Grenzen mit der Maus verschoben werden. Die Dialogbox wird dabei als gepunkteter Rahmen dargestellt. Zu diesem Zweck stellt das AES schon eine Routine *graf_dragbox()* zur Verfügung, die alle Bereichsüberschreitungen abfängt. Diese Routine behält solange die Kontrolle, wie die linke Maustaste gedrückt bleibt. Nach Loslassen dieser Taste wird der Dialogbaum an der ursprünglichen Stelle gelöscht. Jetzt kann die neue Position (*dialbox->ob_x* und *dialbox->ob_y*) berechnet werden. An dieser neuen Position werden wieder die Vorbereitungen zum Zeichnen der Box getroffen (also Hintergrundspeicher reserviert). Hierauf wird die Maus auf den Pfeil (*ARROW*) umgeschaltet und die Mauskontrolle wieder abgegeben [*wind_update()*].

Nachdem jetzt alle zur Dialogabwicklung notwendigen Routinen vorgestellt wurden, müssen diese nur noch in koordinierter Reihenfolge aufgerufen werden. Zur Abwicklung eines Standard-Dialogs dient die Routine *do_dial()*, die nach Beendigung eines Dialogs den Index des Objekts zurückliefert, welches zum Verlassen der Dialogbox benutzt wurde. Übergeben wird der Zeiger auf die Objektstruktur des gewünschten Objektbaums. Beim Abarbeiten von *do_dial()* wird als erstes die Routine *vor_dial()* aufgerufen, die ja bekannterweise (siehe oben) die Vorbereitungen zur Dialogboxdarstellung trifft. Jetzt kann durch *draw_dial()* der Dialogbaum gezeichnet werden. Als *dialtype* [der zweite Parameter der ..._-*dial()*-Aufrufe] muß immer eine Zahl ungleich 1 und 2 (hier 0) angegeben werden. Nachdem der Dialog jetzt auf dem Bildschirm steht, kann das AES die Verwaltung übernehmen [*form_do()*]. Nach Beendigung des Dialogs wird das eventuell selektierte Exit-Objekt wieder deselektiert und der Dialog vom Bildschirm entfernt [*nach_dial()*]. Genauso wie die vorgenannte Routine *do_dial()* ist auch die Prozedur für die Pop-Up-Dialoge *do_PopUp()* aufgebaut. Einzig der andere Dialogtyp (*dialtype=2*) wird verwandt. Etwas aufwendiger ist die für verschiebbare Dialogboxen zuständige Routine *do_movedial()* gestaltet. Nach der bekannten Vorbereitung wird solange in einer Schleife (do..while) verblieben, bis ein Exit-Objekt

```
                  a.g_y = dialbox->ob_y;
149:              a.g_w = dialbox->ob_width;
                  a.g_h = dialbox->ob_height;
150:              break;
151:      default:/* STANDARDDIAL */
152:              /* Dialogbox zentrieren */
153:              form_center(dialbox,&a.g_x,&a.g_y,
                              &a.g_w,&a.g_h);
154:              break;
155:    }
156:
157:    /* Dialogbox ausgeben */
158:    objc_draw(dialbox,ROOT,MAX_DEPTH,a.g_x,a.g_y,
                  a.g_w,a.g_h);
159: }
160:
161: /*+----------------------------------------+*/
162: /*| Nachbereitung der Dialogbox-Darstellung |*/
163: /*+----------------------------------------+*/
164: void nach_dial(OBJECT *dialbox, int dialtype)
165: {
166:   GRECT a;
167:
168:   switch (dialtype)
169:   {
170:     case 1: /* MOVEDIAL */
171:             a.g_x = dialbox->ob_x;
                a.g_y = dialbox->ob_y;
172:             a.g_w = dialbox->ob_width;
                a.g_h = dialbox->ob_height;
173:             break;
174:     case 2: /* POPUPDIAL */
175:             a.g_x = dialbox->ob_x;
                a.g_y = dialbox->ob_y;
176:             a.g_w = dialbox->ob_width;
                a.g_h = dialbox->ob_height;
177:             break;
178:     default:/* STANDARDDIAL */
179:             /*   Dialogbox zentrieren */
180:             form_center(dialbox,&a.g_x,&a.g_y,
                             &a.g_w,&a.g_h);
181:             break;
182:   }
183:
184:   /*   Hintergrund freigeben */
185:   form_dial(FMD_FINISH, a.g_x,a.g_y,a.g_w,a.g_h,
                a.g_x,a.g_y,a.g_w,a.g_h);
186:   /*   Erneuerung beendet */
187:   wind_update(END_UPDATE);
188: }
189:
190: /*+------------------------------------------+*/
191: /*| 'MOVEDIAL'-Box auf Desktop verschieben    |*/
192: /*+------------------------------------------+*/
193: void movedial(OBJECT *dialbox, int dialtype)
194: {
195:   int nx,ny;
196:   GRECT desk;
197:
198:   if (dialtype == 1)  /* es ist also eine
                              'MOVEDIAL'-Box */
199:   {
200:     wind_update(BEG_UPDATE);
                              /* Fenstererneuerung */
201:     wind_update(BEG_MCTRL);   /* Mauskontrolle */
202:     graf_mouse(FLAT_HAND,0);  /* Maus auf
                                   "flache Hand" */
203:
204:     /* GröPe des Desktop ermitteln */
205:     wind_get(0,WF_WORKXYWH,&desk.g_x,&desk.g_y,
                  &desk.g_w,&desk.g_h);
206:
207:     /* Rechteck an neue Position verschieben */
208:     graf_dragbox(dialbox->ob_width,
                      dialbox->ob_height,
209:                  dialbox->ob_x,
                      dialbox->ob_y,
210:                  desk.g_x,         desk.g_y,
211:                  desk.g_w,         desk.g_h,
212:                  &nx,&ny);
213:
214:     /* Alte Object-Zeichnung löschen (Speicher
           freigeben) */
215:     form_dial(FMD_FINISH, dialbox->ob_x,
                   dialbox->ob_y,
216:               dialbox->ob_width,
                   dialbox->ob_height,
```


MGE
maxon graphic
expansion

Die Grafikerweiterung für Profis

Vorbei sind die Tage der Eintönigkeit, die MAXON Graphic Expansion bringt Farbe ins Spiel: Gleichzeitig bis zu 256 Farben oder echte Graustufen, Auflösungen von bis zu 1664*1200 Bildpunkten, blitzschneller Bildaufbau durch einen leistungsfähigen Grafikprozessor - dies sind die Merkmale, die die MGE zur idealen Grafikerweiterung für Profis machen. Speziell ausgerichtet auf die Anforderungen kreativer Anwendungen wie DTP, CAD oder elektronischer Bildverarbeitung eröffnet sie durch ihre enorme Flexibilität neue Dimensionen der Produktivität und verwandelt den Mega ST in eine Grafik-Workstation.

Intelligenz und Stärke:

INTELs Grafikprozessor 82786 treibt die MGE zu grafischen Höchstleistungen. Die serienmäßige CLUT stellt eine Palette von 16,7 Millionen Farben zur Verfügung. Die flexible Programmierbarkeit erlaubt beliebige Bildwechselfrequenzen. Ein freier Sockel lädt zum Einsatz eines mathematischen Coprozessors ein. Die Fähigkeit zur externen Synchronisation ermöglicht den Einsatz der MGE in Video- oder Genlock-Anwendungen. Und in ihrem Drang, bis zu 2 Millionen Bildpunkte gleichzeitig darzustellen, wird die MGE meist nur vom angeschlossenen Monitor zurückgehalten.

Einsetzen und loslegen:

Durch die GDOS-VDI-Treiber sind GEM-Programme auf der MGE lauffähig. Für nonkonforme Software wird eine S/W-Großbildschirmemulation mitgeliefert. VDI-Druckertreiber und Monitorbibliotheken sorgen sich um Routinejobs. Das Kontrollfeld ermöglicht die freie Einstellung von Auflösung und Farben. Und mit den mitgelieferten Bindings für Turbo-C, Assembler und GFA-BASIC wird jeder Programmierer zum Herrn der Formen, Farben und Frequenzen.

Geprüft und für gut befunden:

So urteilt die Fachpresse über die MGE. Etwa c't 5/90: "Die konzeptionelle und praktisch realisierte Vielseitigkeit dieser Karte ... basiert auf den exzellenten Eigenschaften des eingesetzten Grafikprozessors." Ebenso COMPUTER LIVE 8/90: "Für alle Atari-Besitzer, die sich auf professionellem Niveau mit DTP, CAD oder Desktop-Video beschäftigen wollen, bietet MAXON mit der MGE II eine schnelle, extrem vielseitige und leistungsstarke Lösung." Oder TOS 6/90: "Ein Eldorado für alle programmierenden Grafik-Enthusiasten..." Auch PAGE 6/90: "Hier hebt sich positiv die MGE II von MAXON Computer hervor..." Und schließlich ST Magazin 8/90: "Fazit: Grafikkarte für professionelle DTP- und CAD-Anwendungen." Da bleibt eigentlich nur die Frage offen, was die Tester wohl zur neuen Version mit der serienmäßigen Palette von 16,7 Millionen Farben gesagt hätten.

Daten und Features:

Grafikprozessor: Intel 82768
Bildspeicher: 1024 KByte
Farbpalette: 16,7 Millionen
gleichzeitig darstellbare Farben/Graustufen: 256
Pixelfrequenzen: 13.75, 27.5, 55 und 110 MHz
Betriebsarten: Interlace/Non-Interlace

Typische Auflösungen:
Interlaced (Halbbildfrequenz):
- 1664 x 1200 Pixel monochrom, 88 Hz
- 1280 x 1024 Pixel in 16 Farben, 65 Hz
- 896 x 684 Pixel in 256 Farben, 66 Hz

Non-Interlaced (Vollbildfrequenz):
- 1280 x 960 Pixel monochrom, 65 Hz
- 896 x 688 Pixel, in 16 Farben, 67 Hz
- 640 x 480 Pixel in 256 Farben, 66 Hz

Besonderheiten: Anschlußmöglichkeit für Genlock oder externe Synchronisation, Sokkel für mathematischen Coprozessor 68881

Software: Installationsprogramm, Treiber für Hardware, GDOS-VDI und Drucker, Monitorbibliotheken, Kontrollfeld als Accessory, Diverse Demos und Bilder, Programmierbibliotheken.

Preise und Systeme:
MGE II .... DM [illegible]
MGE II mit Eizo 9070S (16" Farbmonitor) .... DM [illegible]
MGE II mit Eizo 6500 (21" Graustufengroßbildschirm) DM 6298.-
MGE II mit Eizo 9400 (20" Farbgroßbildschirm) .... DM [illegible]

Sehen und staunen:

Gerne führen Ihnen die folgenden Händler die MGE vor.

Ernst Brinkmann KG
Spitalerstr. 10
2000 Hamburg 1
Tel. 040/30040

Sienknecht
Bürokommunikation
Heiligengeiststr. 20
2120 Lüneburg
Tel. 04131/46122

MCC
Holzkoppelweg 19a
2300 Kiel 1
Tel. 0431/54381

PS-DATA
Hard- und Software GmbH
Faulenstr. 48-52
2800 Bremen
Tel. 0421/170577

Kurt Neumann PC
Bürger 160
2850 Bremerhaven
Tel. 0471/42006

COM-DATA GmbH
Schiffgraben 19
3000 Hannover 1
Tel. 0511/326736

3K
Computerbild GmbH
Sassenfeld 71
4054 Nettetal 1
Tel. 02153/60001

CSA Computersysteme
Hüttenstr. 56
4650 Gelsenkirchen
Tel. 0209/203420

Data Fach
GmbH & Co KG
Neuhausstr. 4
5100 Aachen
Tel. 0241/4778537

Werbestudio Abakus
Römerstr. 24
5300 Bonn 1
Tel.: 0228/635712

Lehr Bürocenter GmbH
Güterstr. 82
5500 Trier
Tel. 0651/209710

Eickmann Computer
In der Römerstadt 249
6000 Frankfurt 90
Tel. 069/763409

Hard- & Softwareservice
Christine Veigl
Adalbertstr. 61-63
6000 Frankfurt 90
Tel. 069/774043

Pauly Büromaschinen
Salzgasse 6
6250 Limburg/Lahn
Tel. 06431/50040

WAVE
Computersysteme
Südanlage 20
6300 Gießen
Tel. 0641/72357

Schreiber Computer
Rotebühlplatz 10
7000 Stuttgart 1
Tel. 0711/221996

Walliser & Co KG
Marktstr. 48
7000 Stuttgart 50
Tel. 0711/567143

Comp & Phone
Alleenstr. 66
7312 Kirchheim-Teck
Tel. 07021/3949

GCS Rössler
Johnstr. 1
7343 Kuchen
Tel. 07331/81925

Besch & Partner
Stuttgarter Str. 58
7432 Bad Urach
Tel. 07125/8199

Erhardt Bürotechnik
Am Ludwigplatz
7500 Karlsruhe 1
Tel. 0721/16080

Jöst Computer
Im Wendelrot 5
7520 Bruchsal
Tel. 07251/103091

Computer Ludwig
City Studio
Am Rindermarkt 6
8000 München 2
Tel. 089/2609801

JOBI'S Unternehmenssupport
DV & Marketing
Fliegenstr. 12
8000 München 2
Tel. 089/2608703

Computer Corner
Landshuter Str. 4
8313 Vilsbiburg
Tel. 08741/421

HB Computer
DTP-Center
Äußere
Bayreuther 57a-59
8500 Nürnberg 10
Tel. 0911/995140

CCN Computer Center
Nürnberg GmbH
Gibitzenhofstr. 86
8500 Nürnberg 70
Tel. 0911/421056

Sekul & Partner GmbH
Karmelitenstr. 26
8700 Würzburg
Tel. 0931/57555

V. Willgerodt
Bürobedarfshaus
GmbH & Co KG
Hanauer Str. 12
8750 Aschaffenburg
06021/21375

Adolf & Schmoll
Computer
Schwalbenstr. 1
8900 Augsburg
Tel. 0821/528533

Händleranfragen erwünscht!

MAXON Computer • Schwalbacher Str. 52 • 6236 Eschborn
Tel.: 0 61 96/4818 11 • FAX: 06196/41885

ungleich dem *Drag*-Balken (mit dem Index *DRAGGER*) zum Verlassen der Dialogbox angeklickt wurde. Innerhalb der Schleife wird der Dialog gezeichnet [*draw_dial()*] und, falls das *DRAGGER*-Objekt gewählt wurde, durch Aufruf von *move_dial()* verschoben.

Die vorgestellten *do_...()*-Routinen sind so natürlich nur für Boxen mit einem Exit-Button zu gebrauchen, die nicht weiter vom Programm verwaltet werden müssen. Das bedeutet, daß sie bei Verwendung von programmverwalteten Dialogen (z.B. Hoch- und Herunterzählen eines Wertes durch zwei Buttons) noch erweitert werden müssen. Eine solche Erweiterung durch einen *do{...}while !ende*-Block ist im Beispielprogramm Listing 4 und 5 dargestellt. Bevor wir dieses Beispiel näher betrachten, sollen die *MOVEDIAL-Library* erstmal compiliert und die entsprechenden Dateien an den richtigen Platz kopiert werden. Zum Compilieren der Library muß nur die Option *Compile...* des Turbo C-Entwicklungspakets benutzt werden. Das hierdurch entstehende Objekt-File muß nicht gelinkt werden. Sind keine Fehler (oder Warnungen) aufgetreten (.H-File Listing 2 muß beim Compilieren im gleichen Ordner wie der Quelltext Listing 1 stehen!), können die einzelnen Files an ihren endgültigen Platz gebracht werden. Das .H-File muß dorthin, wo der Compiler seine Include-Dateien sucht. Das compilierte Objekt-File (.O) der Library muß in den Ordner, aus dem der Linker seine Libraries zusammensucht. Nun ist es möglich, das Beispielprogramm Listing 4 einzugeben, zu compilieren und zu linken. Damit alles etwas einfacher abläuft, kann die Projekt-Funktion des Turbo C-Compilers benutzt werden. Als Projekt-Datei sollte dabei Listing 3 eingesetzt werden. Jetzt kann zum Compilieren, Linken und Starten des Beispielprogramms (oder eigener, die *MOVEDIAL-Library* benutzender Programme) die *RUN...*-Funktion des Turbo C benutzt werden. In eigenen Programmen muß vor Benutzung von *MOVEDIAL-Routinen* die Prototypen-Deklaration (Listing 2), also das .H-File included werden.

Beispielprogramm

Da in dem Beispielprogramm Listing 4 eine Resource-Datei benötigt wird, soll deren Aufbau hier aufgezeigt werden. Die Abbildungen 4, 5, 6 und 7 zeigen die vier benötigten Dialogboxen, um alle Features der Library zu testen. Das genaue Aussehen ist nicht so wichtig. Einzig die durch serifenlose Schrift gekennzeichneten Buttons sollten auch genau so definiert und mit genau diesem Namen versehen werden.

```
217:                 dialbox->ob_x, dialbox->ob_y,
218:                 dialbox->ob_width,
                     dialbox->ob_height);
219:
220:      /* Jetzt die Verschiebung in die OBJECT-
             Struktur eintragen */
221:      dialbox->ob_x += nx-dialbox->ob_x;
222:      dialbox->ob_y += ny-dialbox->ob_y;
223:
224:      /* Speicher an der neuen Position
             reservieren */
225:      form_dial(FMD_START, dialbox->ob_x,
                     dialbox->ob_y,
226:                 dialbox->ob_width,
                     dialbox->ob_height,
227:                 dialbox->ob_x,dialbox->ob_y,
228:                 dialbox->ob_width,
                     dialbox->ob_height);
229:
230:      graf_mouse(ARROW,0);      /* Maus wieder
                                       Pfeil */
231:      wind_update(END_MCTRL);   /* keine
                                       Mauskontrolle */
232:      wind_update(END_UPDATE);  /* Erneuerung
                                       beendet */
233:    }
234:  }
235:
236:  /*+-----------------------------------------+*/
237:  /*| 'STANDARD' Dialog erledigen              |*/
238:  /*+-----------------------------------------+*/
239:  int do_dial(OBJECT *tree)
240:  {
241:    int ret;
242:
243:    /* Vorbereitung */
244:    vor_dial(tree,0);
245:
246:    /* Durchführung */
247:    draw_dial(tree,0);
248:
249:    ret = form_do(tree,0); /* dies erledigt
                                  das AES */
250:    tree[ret].ob_state &= ~SELECTED;
                    /* SELECTED zurücksetzen */
251:
252:    /* Nachbereitung */
253:    nach_dial(tree,0);
254:
255:    return(ret);
256:  }
257:
258:  /*+-----------------------------------------+*/
259:  /*| 'MOVEDIAL' Dialog erledigen              |*/
260:  /*+-----------------------------------------+*/
261:  int do_movedial(OBJECT *tree, int DRAGGER)
262:  {
263:    int ret;
264:
265:    /* Vorbereitung */
266:    vor_dial(tree,1);
267:
268:    /* Durchführung */
269:    do
270:    {
271:      /* Dialogbox ausgeben*/
272:      draw_dial(tree,1);
273:
274:      ret = form_do(tree,0);
          /* dies erledigt das AES */
275:      tree[ret].ob_state &= ~SELECTED;
          /* SELECTED zurücksetzen */
276:
277:      if (ret==DRAGGER)
278:        movedial(tree,1);
279:
280:    } while (ret == DRAGGER);
281:
282:    /* Nachbereitung */
283:    nach_dial(tree,1);
284:
285:    return(ret);
286:  }
287:
288:  /*+-----------------------------------------+*/
289:  /*| 'POPUP' Dialog erledigen                 |*/
290:  /*+-----------------------------------------+*/
```

→

Alles andere ist nur schmückendes Beiwerk, um die Funktion der Dialogbox zu testen. Wichtig ist auch noch der ebenfalls mit serifenloser Schrift angegebene Name der jeweiligen Dialogbox. Er sollte genau so eingegeben werden, um Fehlermeckern des Compilers zu verhindern. Die angegebenen Dialogboxen kann nun jeder selbst mit dem Resource-Construction-Set (jedem Turbo C-Paket beiliegend) eingeben und abspeichern (in den gleichen Ordner wie der Beispielquelltext). Als Name der Datei sollte ebenfalls *MOVEDIAL* (wie auch für den Beispielquelltext) angegeben werden. Wurde alles richtig gemacht, kann das Beispielprogramm compiliert, gelinkt und gestartet werden. Daraufhin erscheint das Pop-Up-Menü (Abbildung 5), von dem aus die anderen Menüs erreicht werden können. Wird *MOVEDIAL* gewählt, erscheint die *MOVEDIAL*-Box (Abbildung 4) in der Bildschirmmitte. Die Dialogbox kann nun durch Festhalten des *Dragger*-Balkens auf dem Desktop verschoben werden. Nach Verlassen der Box erscheint wieder das Pop-Up-Menü. Bei erneutem Aufruf der *MOVEDIAL*-Box taucht diese wieder an dem Platz auf, wo sie zuletzt stand. Die anderen Dialogboxtypen sind ähnlich aufrufbar und entsprechend ihrer Möglichkeiten zu benutzen.

In Listing 4 soll nun die Routine *do_PopUp_menue()* etwas genauer betrachtet werden. Da dort die Hauptdialogbox (ein Pop-Up-Dialog) des Beispielprogramms verwaltet wird, werden als erstes die Startadressen aller benutzten Dialogboxen (Objektbäume) ermittelt [durch *rsrc_gaddr()*]. Danach wird ein do..while-Konstrukt solange durchlaufen, bis der *QUIT*-Button betätigt wird. Innerhalb der Schleife werden als erstes die Vorbereitungen zum Zeichnen des Hauptdialogs getroffen. Danach wird dieser gezeichnet. Benutzerreaktionen werden wie bekannt durch die AES-Routine *form_do()* abgewickelt, die dann den Index des Objekts zurückliefert, das zum Verlassen des Dialogs geführt hat. Jetzt kann der Dialog wieder vom Bildschirm entfernt werden. Wurde ein Objekt ungleich dem *QUIT*-Button betätigt, so werden die entsprechenden Dialogboxen aufgerufen (innerhalb des *switch()*-Konstrukts). Hier können selbstverständlich auch eigene Unterprogramme aufgerufen werden. Würde man aber auf diese Weise das oben angeschnittene Problem des in-/dekrementieren eines Wertes über Pfeilbuttons erledigen, wäre dies mehr als unelegant, da jedesmal die komplette Box neu gezeichnet werden müßte. Dazu kann Listing 5 betrachtet werden, das ein Programmfragment darstellt, in dem die notwendige Hülle für ein solches Problem gezeigt ist.

```
291:  int do_popup(OBJECT *tree)
292:  {
293:    int ret;
294:
295:    /* Vorbereitung */
296:    vor_dial(tree,2);
297:
298:    /* Durchführung */
299:    draw_dial(tree,2);
300:
301:    ret = form_do(tree,0);  /* dies erledigt
                                   das AES */
302:    tree[ret].ob_state &= ~SELECTED;
        /* SELECTED zurücksetzen */
303:
304:    /* Nachbereitung */
305:    nach_dial(tree,2);
306:
307:    return(ret);
308:  }
```

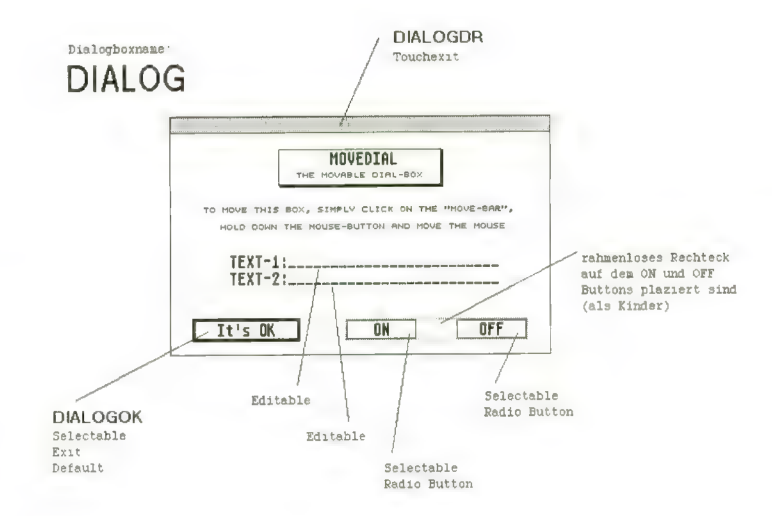


Abb. 4

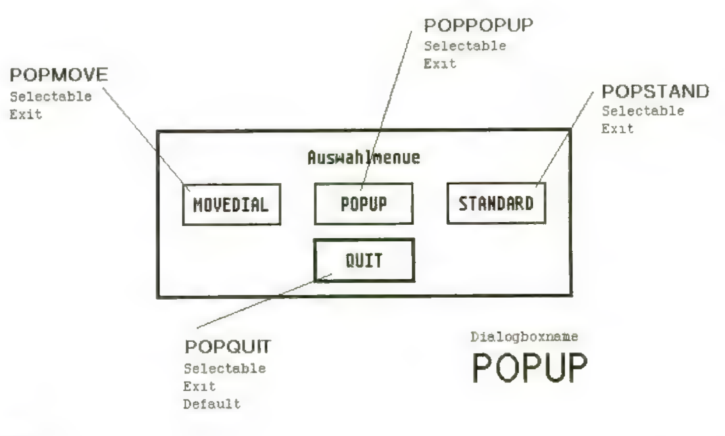


Abb. 5

Erweiterungsmöglichkeiten

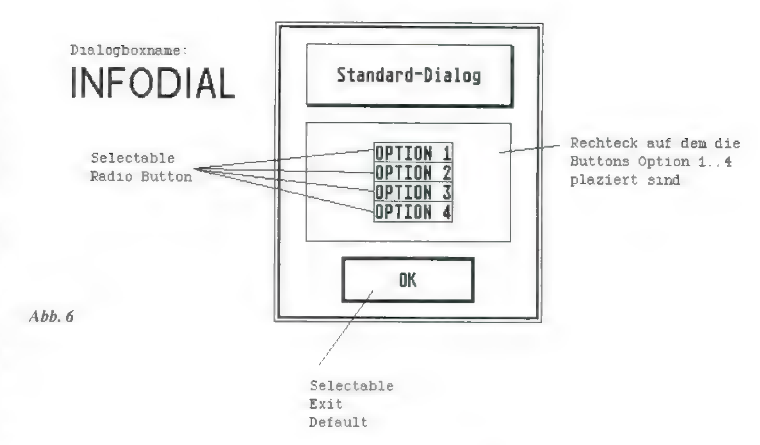


Abb. 6

Dialogboxname
POPPIE
BUTTON 1
BUTTON 2
BUTTON 3
BUTTON 4
Selectable
Exit

Abb. 7

Zum Abschluß möchte ich auch noch die Programmierer unter den Lesern zum Verfeinern der vorgestellten Library anregen. Unter dem Motto „schöner, größer, besser, mehr" läßt sich noch einiges an den Routinen machen.

Als erstes ist hierbei an eine bessere Restaurierung des Hintergrunds zu denken. Zwar wird nach Beendigung des Dialogs die Dialogbox wieder vom Bildschirm entfernt, aber leider bleiben Reste in Fenstern zurück. Diese Rückstände bleiben solange erhalten, bis die Fensterrestaurierungsroutine des Programms aufgerufen werden kann. Ein solcher Aufruf erfolgt normalerweise innerhalb einer *event_multi()*-Prozedur. Wird also eine Dialogbox verlassen, bekommt das AES eine *REDRAW-Message*, was das eigene Programm dazu veranlaßt, die entsprechenden Fenster zu restaurieren. Dies funktioniert so natürlich nicht bei den *MOVEDIAL*-Boxen, da ja auch nach dem Verschieben innerhalb der *do_movedial()*-Prozedur verblieben wird. Wird also eine *MOVEDIAL*-Box auf dem Schirm verschoben, bleiben innerhalb von Fenstern immer Reste zurück. Dem kann man abhelfen (und dies ist der erste Erweiterungsvorschlag), wenn man vor dem Zeichnen [durch *draw_dial()*] den dadurch belegten Bilduntergrund retten würde (in einen Pufferspeicher) und ihn nach Aufruf von *nach_dial()* wieder dorthin zurückkopieren würde. Die benötigten Ausmaße werden ja auch zum Reservieren des Hintergrunds in *vor_dial()* und *nach_dial()* ermittelt. Bleibt nur noch, auf die Größe des Pufferspeichers hinzuweisen. Die benötigte Größe sollte jeweils aus der aktuellen Auflösung berechnet werden, da bei späteren Rechnerversionen (z.B. TT etc.) die Dialogboxen eventuell größer als die heutige Standardauflösung von 640x400 werden könnten.

Eine schöne Sache wäre auch das Verschieben der kompletten Box statt nur eines Rahmens. Dazu kann die vorgenannte Routine benutzt werden. Nur muß hier auf eine Veränderung der Mausposition gewartet und jedesmal der Bildschirm restauriert und die Box neu gezeichnet werden. Damit das Zeichnen schneller vonstatten geht, wird die Box beim Verschieben innerhalb des Bildschirms kopiert, also nicht durch *draw_dial()* neu gezeichnet. Um das ganze flimmerfrei erledigen zu können, ist folgender Ablauf denkbar:

A Bildschirmhintergrund komplett retten
B Dialogbox ein erstes Mal zeichnen [durch *draw_dial()*]
C beim Verschieben zuerst die Box an die neue Position kopieren und nur die benötigten Teile des Originaluntergrunds wiederherstellen (also die Differenz zwischen der neuen und der alten Position)

Leider kann diese flimmerfreie Methode schon einiges an Speicherplatz für den Bildpuffer kosten (man denke nur an Großmonitore mit 1024x768 oder mehr Pixeln Auflösung).

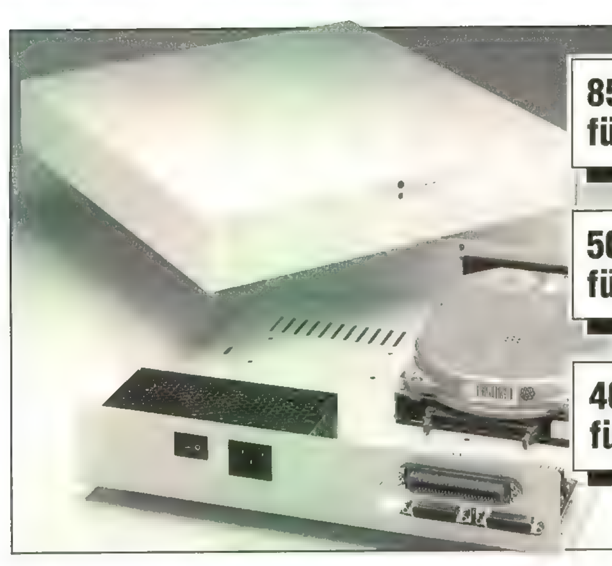
Ein letzter Tip betrifft das Wiederherstellen einer einmal gemachten Positionierung von MOVEDIAL-Boxen auf dem Bildschirm. Da nach jedem Neustart eines Programms mit MOVEDIAL-Boxen diese insgesamt neu positioniert werden müssen, wäre eine Methode vorteilhaft, die alle Dialogbox-Positionen wieder herstellen kann. Dazu müßten auf Wunsch alle Positionen in einer Datei abgelegt und bei Programmstart wieder eingelesen werden können (ähnlich *DESKTOP.INF*). Dabei muß allerdings jeweils noch eine Auflösungsanpassung vorgenommen werden. Wenn man sich vorstellt, daß beim Abspeichern der Positionen ein Monitor mit einer Auflösung von 1024x768 Pixeln verwandt wurde und einige Boxen am unteren Bildschirmrand positioniert waren, wird klar, daß ein Wiederherstellen dieser Dialogboxpositionen bei einer 640x400-Pixel-Auflösung schiefgehen muß. Man könnte die Auflösung, die beim Abspeichern eingestellt war, mit in die Info-Datei ablegen, um so die Dialogboxpositionen auf die aktuelle Auflösung bei erneutem Programmstart rückrechnen zu können.

Ein Wort soll noch an die Programmierer anderer Programmiersprachen gerichtet werden. Wer nicht über einen C-Compiler verfügt, kann das Programm auf alle Programmiersprachen umschreiben, in denen Zugriff auf die Objektstrukturen besteht. Als Beispiel hierfür soll nur Pascal genannt werden. Sicherlich sind auch alle Modula-2-Implementationen hierfür geeignet.

Ich hoffe, daß die vorgestellten Routinen einen tieferen Einblick in die Objektprogrammierung gestatten und sie in eigenen Programmentwicklungen recht zahlreich (und erfolgreich) eingesetzt werden.

Matthias Baldauf

SCSI-Festplatten zu »Schotten-Preisen«!



Zum Beispiel:

85 MB SCSI-Festplatte (28 ms) für nur DM 1.198,–

50 MB SCSI-Festplatte (28 ms) für nur DM 1.098,–

40 MB SCSI-Festplatte (19 ms) für nur DM 1.198,–

Unsere SCSI-Festplatten werden komplett anschlußfertig incl. Software und Kabel ausgeliefert.

Ausstattung und Leistungsmerkmale unserer Festplatten:

- Datentransferraten >600 KByte/s (mit CDC- und Maxtorlaufwerken bis zu 850 KByte/s erzielbar), mittlere Zugriffszeiten bis zu 14 ms
- Spitzensoftware: 255 Partitionen installierbar, Passwortfunktion, jede Partition autobootfähig, Interleave 1:1 einstellbar, Cache, Backup, Optimizer in der Software enthalten
- 100% Atari-kompatibel, sämtliche Fremdbetriebssysteme (PC-Speed, PC-Ditto, Spectre, Aladin, Minix, OS-9, RTOS) sind voll lauffähig
- Superleise (3,5"-Festplatten ohne Lüfter, 5,25"-Festplatten mit thermogeregeltem Lüfter)
- Durchgeschleifter gepufferter DMA-Bus, Autoparkfunktion hardwaremäßig
- Herausgeführter SCSI-Bus (50-poliger Centronics-Anschluß, Apple MacIntosh und PC's anschließbar)
- Zweite SCSI-Festplatte im Gehäuse nachrüstbar (SCSI-Hostadapter und Gehäuse für interne zweite Festplatte vorbereitet)
- Unsere SCSI-Festplatten werden komplett anschlußfertig im Gehäuse incl. Netz-, DMA-Kabel, Software und Handbuch geliefert

Preise:

Modell	Preis
32 MB, 40 ms, ST138N-0	DM 998,–
40 MB, 19 ms, Quantum	DM 1.198,–
49 MB, 28 ms, ST157N-1	DM 1.098,–
85 MB, 28 ms, ST296N	DM 1.198,–
80 MB, 24 ms, ST1096N	DM 1.298,–
80 MB, 19 ms, Quantum	DM 1.798,–
170 MB, 28 ms, 2x ST296N	DM 2.498,–
280 MB, 17 ms, Maxtor	DM 3.498,–
380 MB, 17 ms, Maxtor	DM 3.998,–
702 MB, 14 ms, CDC	DM 5.998,–
1200 MB, 14 ms, CDC	DM 11.998,–
44 MB, 25 ms, SQ 555	DM 1.598,–

SCSI-Kits (Festplatte und SCSI-Hostadapter für ST):

Kit	Preis
32 MB Kit (ST138N-0)	DM 798,–
40 MB Kit (P40S)	DM 998,–
49 MB Kit (ST157N-1)	DM 898,–
85 MB Kit (ST296N)	DM 998,–
80 MB Kit (ST1096N)	DM 1.098,–
80 MB Kit (P80S)	DM 1.598,–

Zubehör	Preis
SCSI-Hostadapter (incl. Software und DMA-Kabel)	DM 198,–
DMA-Kabel	DM 39,–
SCSI-Kabel	DM 39,–
Netzteil 50 W	DM 99,–
Gehäuse	DM 99,–
Cartridge für SQ555	DM 239,–

Weitere Modelle sowie sonstige Soft- und Hardware auf Anfrage!



Datensysteme

Eugenstraße 28
7302 Ostfildern 4

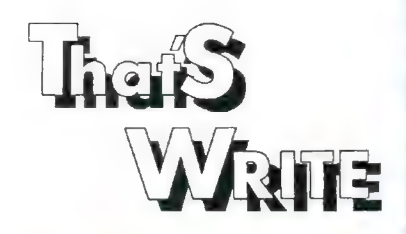
Telefon 0711/4579623
Telefax 0711/4569566

```
 1:  /***************************************/
 2:  /*            L I S T I N G  2         */
 3:  /***************************************/
 4:  /*  MOVEDIAL-Library für Dialogboxen   */
 5:  /*  written in Turbo C 2.0             */
 6:  /*-------------------------------------*/
 7:  /*  Include-File: MOVELIB.H            */
 8:  /*-------------------------------------*/
 9:  /*  by M.Baldauf 6/1990                */
10:  /*  (c) MAXON Computer                 */
11:  /***************************************/
12:
13:  #include <aes.h>
14:  #include <vdi.h>
15:
16:  int gem_init(void);
17:
18:  void gem_exit(int handle);
19:
20:  void vor_dial(OBJECT *dialbox, int dialtype);
21:
22:  void draw_dial(OBJECT *dialbox, int dialtype);
23:
24:  void nach_dial(OBJECT *dialbox, int dialtype);
25:
26:  void movedial(OBJECT *dialbox, int dialtype);
27:
28:  int do_dial(OBJECT *tree);
29:
30:  int do_movedial(OBJECT *tree, int DRAGGER);
31:
32:  int do_popup(OBJECT *tree);
```

```
 1:  /***************************************/
 2:  /*            L I S T I N G  3         */
 3:  /***************************************/
 4:  /*  MOVEDIAL-Library für Dialogboxen   */
 5:  /*  written in Turbo C 2.0             */
 6:  /*-------------------------------------*/
 7:  /*  Projekt-Datei: MOVELIB.PRJ         */
 8:  /*-------------------------------------*/
 9:  /*  by M.Baldauf 6/1990                */
10:  /*  (c) MAXON Computer                 */
11:  /***************************************/
12:
13:  *                    ; name of executable program
                            is topmost window
14:  =                    ; list of modules follows...
15:
16:  TCSTART.O            ; startup code
17:
18:  *                    ; compile topmost window
19:
20:  MOVELIB.O            ; unsere neue MOVELIB muP
                            auch dazugelinkt werden!
21:  TCSTDLIB.LIB         ; standard library
22:  TCEXTLIB.LIB         ; extended library
23:  TCTOSLIB.LIB         ; TOS library
24:  TCGEMLIB.LIB         ; AES and VDI library
```

```
 1:  /*****************************************/
 2:  /*              L I S T I N G  4         */
 3:  /*****************************************/
 4:  /*  Demoprogramm der MOVEDIAL-Routinen   */
 5:  /*  written in Turbo C 2.0               */
 6:  /*---------------------------------------*/
 7:  /*  Sourcefile: MOVEDIAL.C               */
 8:  /*---------------------------------------*/
 9:  /*  by M.Baldauf 6/1990                  */
10:  /*  (c) MAXON Computer                   */
11:  /*****************************************/
12:
13:  #include "movedial.h"    /* RSC-Definitionen */
14:  #include "movelib.h"     /* Unterprogramm-
                                 Deklarationen */
15:  #include <aes.h>         /* AES-Routinen */
16:  #include <vdi.h>         /* VDI-Routinen */
17:
18:  void do_popup_menue(void);
19:  void do_popup_menue(void)
20:  {
21:    /* Objecte (Dialogboxen, Menü, etc.)
          deklarieren */
22:    OBJECT *dialog,   /* MOVEDIAL-Box */
23:           *infodial,/* STANDARD-Dialogbox */
24:           *poppie,   /* POPUP-Dialogbox */
25:           *popup;    /* Auswahl-(Popup-)Dialog */
26:
27:    int ret;
28:
29:    rsrc_gaddr(R_TREE, DIALOG,    &dialog);
30:    rsrc_gaddr(R_TREE, POPUP,     &popup);
31:    rsrc_gaddr(R_TREE, INFODIAL,  &infodial);
32:    rsrc_gaddr(R_TREE, POPPIE,    &poppie);
33:
34:    /* Durchführung */
35:    do
36:    {
37:      /* Vorbereitung */
38:      vor_dial(popup,2);
39:
40:      /* Dialogbox ausgeben (zeichnen) */
41:      draw_dial(popup,2);
42:
43:      ret = form_do(popup,0); /* dies erledigt
                                    das AES */
44:      popup[ret].ob_state &= ~SELECTED;
         /* SELECTED zurücksetzen */
45:
46:      /* Nachbereitung */
47:      nach_dial(popup,2);
48:
49:      /* was hat der Benutzer gewählt? */
50:      switch (ret)
51:      {
52:        case POPMOVE:   do_movedial(dialog,
                                       DIALOGDR);
53:                        break;
54:        case POPPOPUP:  do_popup(poppie);
55:                        break;
56:        case POPSTAND:  do_dial(infodial);
57:                        break;
58:        default:        break;
59:      }
60:
61:    } while (ret != POPQUIT);
62:  }
63:
64:  int main(void)
65:  {
66:    int handle;
67:    char rsc_name[] = "MOVEDIAL.RSC";
68:
69:    /* Initialisierung der Applikation */
70:    handle = gem_init();
71:
72:    if (handle >= 0)
73:    {
74:      /* Maus als Bienchen */
75:      graf_mouse(HOURGLASS,0L);
76:
77:      /* Resource-Datei laden */
78:      if (rsrc_load(rsc_name))
79:      {
80:        /* und los gehts */
81:        graf_mouse(ARROW,0L);
82:
83:        do_popup_menue();
84:      }
85:      else
86:        return(-1);
87:
88:      /* Abmelden der Applikation */
89:      gem_exit(handle);
90:
91:      return(0);
92:    }
93:    else
94:      return(-1);
```

```
 1:  /*****************************************/
 2:  /*           L I S T I N G  5            */
 3:  /*****************************************/
 4:
 5:  /* Vorbereitung */
 6:  vor_dial(dialog,0); /* 0,1,2 */
 7:
 8:  /* Dialogbox ausgeben (zeichnen) */
 9:  draw_dial(dialog,2);
10:
11:  do
12:  {
13:    ret = form_do(popup,0); /* dies erledigt
                                   das AES */
14:    popup[ret].ob_state &= ~SELECTED;
       /* SELECTED zurücksetzen */
15:
16:    /* was hat der Benutzer gewählt? */
17:    switch (ret)
18:    {
19:      case INKREMENT:  /* Objektstring in Wert
                             umwandeln */
20:                       ...
21:                       /* Wert inkrementieren */
22:                       ...
23:                       /* diesen Wert wieder in
                             String umwandeln */
24:                       ....
25:                       /* in Objektstruktur
                             einpassen */
26:                       ...
27:                       /* und dieses Objekt neu
                             zeichnen */
28:                       ...
29:                       break;
30:      case DEKREMENT:  /* Objektstring in Wert
                             umwandeln */
31:                       ...
32:                       /* Wert dekrementieren */
33:                       ...
34:                       /* diesen Wert wieder in
                             String umwandeln */
35:                       ....
36:                       /* in Objektstruktur
                             einpassen */
37:                       ...
38:                       /* und dieses Objekt neu
                             zeichnen */
39:                       ...
40:                       break;
41:      default:         break;
42:    }
43:
44:  } while ((ret != OK) && (ret != ABBRUCH));
45:
46:  /* Nachbereitung */
47:  nach_dial(popup,2);
```



NEU Version 1.5

Einschreiber

That's Write - Junior

Textverarbeitung für Einsteiger

Neben dem Können der **Junior**-Version ist besonders der günstige Preis eine Entscheidung für alle, die **leicht** und **richtig** einsteigen wollen.

- **Flexibel,** d.h. einfacher Umgang mit einem komfortablen, **preisgünstigen** Textverarbeitungsprogramm. Jederzeit Aufstieg in die Profi-Version möglich
- **Graphikeinbindung**
- **Seitenlayout**
- **Absatzlayout**
- **Silbentrennung** nach deutschen Regeln, erweiterbar durch Eingabe eigener Wörter in unbegrenzter Höhe
- Bis zu 10 verschiedene **Fonts** (Schriftarten) in einem Text möglich
- Lieferung mit **10 Fonts (10 versch. Schriften gleich dabei)**
- **Ausdruck im Graphik- und Text-Modus** (auch gemischt innerhalb einer Zeile)
- **Blockfunktionen**
- **Suchen und ersetzen**
- **Tabulatoren**
- **ASCII einlesen und speichern**
- **Bilddarstellung entspricht dem Ausdruck** (Keine Eingabe von Steuerzeichen nötig)
- **Leicht verständliches deutsches Handbuch**

Preis: DM **148,-**

Aufstieg

Jederzeit **Aufstieg in die Profi-Version** möglich (Original-Diskette der Junior-Version an Heim-Verlag einsenden und Profi-Version bestellen)

Preis: DM **246,-**

Alle genannten Preise sind unverbindlich empfohlene Verkaufspreise.

Kundenbetreuung durch Compo Software GmbH

Informationen und Hotline

Telefon: 0 65 51 - 62 66

Vielschreiber

That's Write

Die professionelle Textverarbeitung

Endlich! Ein Textverarbeitungsprogramm, das neben seinem Können auch anwenderfreundlich ist.

That's Write ist für alle gängigen Anwendungen einer Textverarbeitung geeignet. Dank der einfachen Bedienung per Maus ist es für den Einsteiger leicht und schnell zu erlernen. Dem viel schreibenden Anwender bietet es die Bedienung über die Tastatur oder die Definition eigener Funktionen über Makros.

Komplettpaket mit großem Funktionsumfang:
• eigener Fonteditor • international mehrfach frei belegbare Tastatur • alle Tasten mit Floskeln und Makros belegbar • zuverlässige eingebaute Silbentrennung • umschaltbare Trennregeln für verschiedene Sprachen • eingebaute Rechtschreibkorrektur • das Hauptwörterbuch kann gewechselt werden (Option Englisch, Niederländisch, weitere in Vorbereitung) • eingebaute Serienbrieffunktion mit Schnittstelle zu Datenbanken und Adressverwaltungen • integrierte Schnittstelle zu Accessories, Datenübernahme aus Adressverwaltung • autom. Erstellung eines Stichwortverzeichnises • autom. Erstellen eines Inhaltsverzeichnises • autom. Fußnotenverwaltung • autom. Endnotenverwaltung • Umbenennen, kopieren, löschen von Dateien vom Programm aus • Snapshot-Accessory erlaubt aus anderen Programmen Bilder vom Monitor abzugreifen und diese in That's Write zu verwenden • einfache Installation • Gliederungsfunktion

Optimale Druckerausnutzung:
• 9-Nadel-Drucker (alle Auflösungen) • 24-Nadel-Drucker (alle Auflösungen, auch 360x360 dpi) • Laserdrucker (alle Auflösungen) • Atari-Laserdrucker SLM804 über DMA
• Nur Textdruck bei Typenraddruckern (auch Proportionalschrift in Blocksatz!), verschiedene Typenräder anpaßbar

Ausdruck von Text- und Graphiktextmodus in einer Zeile:
• optimaler Randausgleich • unterstützt interne Schriften des Druckers, wie Proportional, Pica, Elite, schmal, breit, doppelt hoch • unterstützt Grafikfonts freier Größe • unterstützt beliebig viele Download-Fonts in einem Text • unterstützt druckerinterne Vektorfonts (Kyocera u.a.)

Durchdachte Bedienung:
Erstmalig ein übersichtliches und schnelles Bearbeiten von selbst langen Texten. z.B.: „Text zuklappen" versteckt in Sekundenbruchteilen den normalen Text – und zeigt nur noch die Kapitelüberschriften. Dort geht man schnell in das gewünschte Kapitel und „klappt" wieder auf. Noch schneller geht es wohl wirklich nicht mehr

Einzigartiges Druckkonzept:
That's Write druckt Grafiktextmodus und Textmodus gleichzeitig! Wer bietet mehr?

Optimierte Seiteneinteilung beseitigt drei klassische Probleme automatisch:
• Umbruch zwischen Überschrift und Text wird automatisch verhindert
• Leerraum nach Seitenumbruch wird nicht ausgedruckt, d.h. gleichmäßiger oberer Rand
• Einzelne Zeilen eines Absatzes alleine auf einer Seite werden automatisch verhindert.

Bildschirmanzeige entspricht dem Ausdruck (WYSIWYG)

Einfachste Bedienung per Maus, Tastatur oder Makros

Flexible Textgestaltung durch Absatz- und Seitenlayouts

Weitere Vorzüge:
• Unterstützt selbständig den Blitter • läuft unter allen bekannten TOS-Versionen und unter GEM 2.2 • unterstützt Farb- und Monochrombildschirm sowie Großbildschirme • korrekte Behandlung von getrennten Wörtern bei Suchen, Ersetzen und Rechtschreibekorrektur • zwei Texte gleichzeitig bearbeitbar, schnelles Wechseln zwischen den Texten • autom. Sichern während des Schreibens in einem einstellbaren Zeitintervall • Einfügen von Kurz-Datum, Lang-Datum, Seitennummer, Folgeseite per Tastendruck • Textstatistik – Anzahl der Wörter, Zeilen, Seiten, Bilder • Zählfunktion für Wörter (und/oder Wort-Teile) • Ausschneiden/Einfügen von Blöcken mit 4 unabhängigen Puffern • Einfüge- und Überschreibmodus auch bei Proportionalschrift • flexibles ASCII-Laden/Sichern zeilen/absatzweise • Textübernahme von 1st Word/Wordplus. Vorhandene Texte werden mit Fußnoten und allen Attributen wie fett, kursiv, hoch/tief, unterstrichen eingelesen • 10 Marken zum schnellen Anspringen von Textpassagen, die mit dem Text abgespeichert werden • Hilfsfunktionen wie „Wort klein schreiben", „Wort groß schreiben", „1. Buchstabe groß/Rest klein" zur schnellen Korrektur von Tippfehlern. • komfortables Bewegen im Text: zeichenweise, wortweise, zeilenweise, absatzweise, Bildschirm-seitenweise Textseitenweise (zur Kontrolle des Seitenumbruchs), Text Anfang/Ende, Block Anfang/Ende, Seitenanfang
• Druckausgabe auf Druckerport (schnelle Direktansteuerung), Druckerport über Bios, RS232 oder Datei – Bilder können beim Drucken auch weggelassen werden (Probeausdrucke, usw.) • So lassen sich zuerst „gerade" und später „ungerade" Seiten ausdrucken, um Papier komfortabel beidseitig zu bedrucken, ohne jedes Blatt einzeln wenden zu müssen
• Hotline für registrierte Anwender

Preis: DM **348,-**

Änderungen von That's Write 1.3 auf Version 1.5:

1. Rechtschreibkorrektur nach Langenscheidt:
   a) Das deutsche Wörterbuch kennt ca. 3 Millionen Wörter
   b) Trennung nach Wörterbuch möglich, daher erstmalig auch korrekte Trennung von zusammengesetzten Wörtern: „Gas-tank", „Hals-tuch"
   c) Fehler bei Groß-/Kleinschreibung werden, wenn eindeutig, automatisch erkannt und behoben!
   d) Sinnvolle Korrekturvorschläge, so wird z.B. bei „Füsiker" nur das Wort „Physiker" vorgeschlagen.
   e) Einfaches Erweitern und Bearbeiten von Zusatzwörterbüchern und Trennungen.
   f) Zusätzliche Wörterbücher erhältlich: Französisch, Englisch, Amerikanisch, Spanisch, Schwedisch, Schweizer-Deutsch
2. Textverkettung – jetzt kann per Knopfdruck in den folgenden oder den vorigen Text gewechselt werden. Seitennummern und Fußnotennummern werden automatisch übernommen
3. Fonts löschen, Anschlagzählung
4. Paßwortschutz verhindert unberechtigtes Lesen von neugierigen, naseweisen Wesen
5. Textinfo speichert zum Text: Autor, Stichworte, Erstellungsdatum
6. Doppelseitiges Drucken, da linke und rechte Kopftexte (Heftrand kann berücksichtigt werden)

• That's Write Junior
• That's Write Profi
jeweils einschließlich Signum* Font-Konverter

Weitere That's Write Font-Disketten lieferbar

Programmer's Toolbox - Dateien

Teil 6: Weitere rekursive Kommandos



M V ZIMMERMANN

Zum zweiten Mal werden heute Kommandos betrachtet, die über die Fähigkeit zur Argumentexpansion verfügen. In der letzten Folge waren dies LS, CP und MV. Diese Liste wird heute erweitert um:

RM - Löschen von Dateien
RMDIR - Löschen von Verzeichnissen
MKDIR - Erzeugen von Verzeichnissen
CHMOD - Ändern des Dateizugriffsmodus'
TOUCH - Aktualisierung der Modifikationszeit von Dateien

Die Kommandos RM und RMDIR

Name
RM, RMDIR - Löschen von Dateien und Verzeichnissen

Anwendung
RM [-IR] Dateiname...
RMDIR Verzeichnis...

Beschreibung
RM löscht eine oder mehrere Dateien. Besteht dabei ein Schreibschutz für die Dateien, wird zunächst abgefragt, ob die Datei wirklich gelöscht werden soll (Antwort „y").

RMDIR löscht alle angegebenen Verzeichnisse, jedoch nur leere.

Optionen
-I Interaktivmodus. Beim Löschen einer Datei wird zunächst rückgefragt. Löschen erfolgt nur bei Eingabe von „y".
-R Rekursives Löschen des Inhalts eines Verzeichnisses, des Inhalts aller enthaltenen Unterverzeichnisse und des Verzeichnisses selber.

Programmierung
Die Kommandos RM und RMDIR sind in Listing 1.14 und 1.15 programmiert.

Der Aufbau des Kommandos RM ähnelt dabei weitgehend dem der bisher betrachteten rekursiven Kommandos. So lassen sich etwa in der Hauptfunktion *rm* (Zeilen 104-145) Programmstellen zur Optionsanalyse und zur Argumentexpansion finden. Es folgt der Aufruf der bereits vertrauten rekursiven Funktionskaskade zur Abarbeitung der *DIR_DESC_LIST* (Zeilen 139-144). Über *rm_dlist* (Zeilen 79-87) und *rm_dir* (Zeilen 67-77) wird dabei zur Funktion *rm_flist* (Zeilen 38-65) abgestiegen. An dieser Stelle erfolgt das Löschen, eventuell ein rekursives, das auch das Entfernen der betroffenen Verzeichnisse beinhaltet (Zeilen 50-54).

Betrachtet man RMDIR und vergleicht seine Beschreibung mit der von RM, fällt auf, daß RMDIR eigentlich funktional in RM enthalten ist. Entsprechend ist Listing 1.16 - die Implementierung von RMDIR - bis auf Umbenennungen auch nichts weiter als ein „abgespecktes" Listing 1.15. In *rmdir* (Zeilen 83-115) fehlen die Möglichkeiten zur Optionsinterpretation. Darüber hinaus fehlen in *rmdir_flist* (Zeilen 34-50) das Löschen von Dateien und der rekursive Abstieg.

Das Kommando MKDIR

Name
MKDIR - Erzeugen von Verzeichnissen

Anwendung
MKDIR Verzeichnis...

Beschreibung
MKDIR erzeugt Verzeichnisse. Die (unsichtbaren) Standardeinträge „." und „.." innerhalb von Verzeichnis werden dabei automatisch miterzeugt.

Programmierung
Das Kommando MKDIR ist in Listing 1.17 programmiert. MKDIR bildet in gewissem Sinne eine Ausnahme von den übrigen Kommandos dieses Serienteils. Seine Ausnahmestellung resultiert daraus, daß MKDIR kein rekursives Kommando ist. Das Modul EXPAND wird entsprechend nicht benötigt. Wegen der nicht zu leugnenden Symmetrie zum Kommando RMDIR wird MKDIR jedoch trotzdem in der heutigen Folge abgedruckt.

Seine Implementierung hält keine Überraschungen bereit. Entsprechend den Vorgaben unter Beschreibung versucht MKDIR, sämtliche übergebenen Parameter als Verzeichnisse anzulegen. Optionen sind dabei nicht zulässig, weshalb die

Implementierung sehr einfach wird und an dieser Stelle nicht weiter betrachtet werden soll.

Das Kommando CHMOD

Name
CHMOD - Ändern des Zugriffsmodus'

Anwendung
CHMOD [+W] [-W] Dateiname...

Beschreibung
Für die angegebenen Dateinamen wird der Schreibschutz entweder gesetzt (-W) oder zurückgenommen (+W).

Programmierung
Das Kommando CHMOD ist in Listing 1.18 programmiert. Bis auf die Funktion *chmod_flist* (Zeilen 38-57) zeigt es nur Vertrautes. In der Funktion *chmod_flist* wird, entsprechend der Vorgabe (*setrw*), das Read-Only-Flag entweder für die betreffenden Dateien gesetzt oder entfernt (Zeilen 49-53). Dabei erfolgt eine Abbildung auf die GEMDOS-Funktion *Fattrib*.

Das Kommando TOUCH

Name
TOUCH - Aktualisierung der Modifikationszeit von Dateien

Anwendung
TOUCH Dateiname...

Beschreibung
TOUCH setzt den Zeitpunkt der letzten Modifikation aller übergebenen Dateinamen auf die Systemzeit.

Programmierung
Das Kommando TOUCH ist in Listing 1.19 programmiert. Bis auf die Funktion *touch_flist* (Zeilen 35-52) zeigt es nur Vertrautes. In der Funktion *touch_flist* wird für die selektierten Dateien das Modifikationsdatum aktualisiert (Zeilen 45-48). Es erfolgt eine Abbildung auf die ATOM-Funktion *touch*.

Vorausschau

In der nächsten Folge ist es soweit: Die Shell wird implementiert. Dabei wird gezeigt, wie Shell-Variablen, Pipelining und Kommandoprozeduren realisiert werden können. Der erste Block der „Programmer's Toolbox" gelangt damit zu seinem Abschluß.

Dirk Brockhaus

```
  1:  /*
  2:   * Listing 1.15, Datei  : rm.c
  3:   * Programm             : RM - Löschen von Dateien
  4:   * Modifikationsdatum   : 16-Nov-89
  5:   * Abhängigkeiten       : stdio.h, string.h,
                                osbind.h,
  6:   *                        local.h, atom.h, expand.h
  7:   */
  8:
  9:  #include <stdio.h>
 10:  #include <string.h>
 11:  #include <osbind.h>
 12:  #include "local.h"
 13:  #include "atom.h"
 14:  #include "expand.h"
 15:
 16:  /*
 17:   * Funktionen    : rm_flist, rm_dir, rm_dlist
 18:   *
 19:   * Parameter     : rm_flist(path, flist, inter,
                                 rec);
 20:   *                 rm_dir(dir, inter, rec);
 21:   *                 rm_dlist(dlist, inter, rec);
 22:   *                 char           *path;
 23:   *                 FILE_DESC_LIST flist;
 24:   *                 BOOLEAN        inter,
 25:   *                                rec;
 26:   *                 DIR_DESC_ITEM  dir;
 27:   *                 DIR_DESC_LIST  dlist;
 28:   *
 29:   * Aufgabe       :
 30:   *
 31:   * Rekursives Löschen der Dateien und
       Verzeichnisse
 32:   * aus <dlist>, relativ zum Pfad <path>. Gemäß dem
 33:   * Wert von <inter> wird beim Überschreiben von
 34:   * bestehenden Dateien entweder "interaktiv"
       zurück-
 35:   * gefragt oder nicht.
 36:   */
 37:
 38:  void rm_flist(path, list, inter, rec)
 39:  char            *path;
 40:  FILE_DESC_LIST  list;
 41:  BOOLEAN         inter,
 42:                  rec;
 43:  {   char  filepath[100],
 44:             newfilepath[100];
 45:
 46:      while (list != NULL) {
 47:          sprintf(filepath, "%s\\%s", path,
 48:                  list->fdesc.fname);
 49:          if (list->fdesc.fattr & 0x10) {
 50:              if (rec) {
 51:                  rm_flist(filepath, list->down,
 52:                           inter, rec);
 53:                  Ddelete(filepath);
 54:              }
```

```
 55:              else
 56:                  fprintf(stderr,
 57:                          "rm: %s is a directory\n",
 58:                          filepath);
 59:          }
 60:          else if (!arm(filepath, inter))
 61:              fprintf(stderr, "rm: Can't delete
                                  %s\n",
 62:                      filepath);
 63:          list = list->next;
 64:      }
 65:  }
 66:
 67:  void rm_dir(dir, inter, rec)
 68:  DIR_DESC_ITEM dir;
 69:  BOOLEAN       inter,
 70:                rec;
 71:  {   if (strlen(dir.dirpath) == 3 &&
 72:          dir.dirpath[2] == '\\')
 73:          dir.dirpath[2] = 0;
 74:      rm_flist(dir.dirpath, dir.flist, inter, rec);
 75:      if (fisdir(dir.dirpath) && rec)
 76:          Ddelete(dir.dirpath);
 77:  }
 78:
 79:  void rm_dlist(dlist, inter, rec)
 80:  DIR_DESC_LIST dlist;
 81:  BOOLEAN       inter,
 82:                rec;
 83:  {   if (dlist != NULL) {
 84:          rm_dir(*dlist, inter, rec);
 85:          rm_dlist(dlist->next, inter, rec);
 86:      }
 87:  }
 88:
 89:  /*
 90:   * Funktion      : rm
 91:   *
 92:   * Parameter     : ok = rm(argc, argv);
 93:   *                 BOOLEAN ok;
 94:   *                 short   argc;
 95:   *                 char    *argv[];
 96:   *
 97:   * Aufgabe       :
 98:   *
 99:   * Interpretierung der durch <argc> und <argv>
100:   * spezifizierten Parameterliste gemäß den Fest-
101:   * legungen des Kommandos rm.
102:   */
103:
104:  BOOLEAN rm(argc, argv)
105:  short argc;
106:  char  *argv[];
107:  {   DIR_DESC_LIST dlist;
108:      char          path[100];
109:      BOOLEAN       option    = TRUE,
110:                    start     = TRUE,
111:                    inter     = FALSE,
```
→

```
112:                  rec      = FALSE;
113:      short          i;
114:
115:      for (i = 1; i < argc; i++) {
116:          convupper(argv[i]);
117:          if (option && argv[i][0] == '-') {
118:              if (strcmp(argv[i], "-I") == 0)
119:                  inter = TRUE;
120:              else if (strcmp(argv[i], "-R") == 0)
121:                  rec = TRUE;
122:              else
123:                  fprintf(stderr,
124:                     "Unknown option %s\n",
                           argv[i]);
125:          }
126:          else {
127:              option = FALSE;
128:              if (start) {
129:                  dlist = built_dlist(argv[i],
                              FALSE,!rec, rec, 0);
131:                  start = dlist->flist ==
132:                          (FILE_DESC_LIST)ERROR;
133:              }
134:              else
135:                  dlist = expand_dlist(dlist,
                              argv[i],FALSE, !rec,
                              rec, 0);
137:          }
138:      }
139:      if (!start) {
140:          rm_dlist(dlist, inter, rec);
141:          drop_dlist(dlist);
142:          if (inter)
143:              printf("\n");
144:      }
145:  }
146:
147:  void main(argc, argv)
148:  short argc;
149:  char  *argv[];
150:  {   if (!rm(argc, argv))
151:          exit(1);
152:      exit(0);
153:  }
```

```
  1:  /*
  2:   * Listing 1.16, Datei : rmdir.c
  3:   * Programm            : RMDIR - Löschen von
  4:   *                       Unterverzeichnissen
  5:   * Modifikationsdatum  : 16-Nov-89
  6:   * Abhängigkeiten      : stdio.h, string.h,
                               osbind.h,
  7:   *                       local.h, atom.h, expand.h
  8:   */
  9:
 10:  #include <stdio.h>
 11:  #include <string.h>
 12:  #include <osbind.h>
 13:  #include "local.h"
 14:  #include "atom.h"
 15:  #include "expand.h"
 16:
 17:  /*
 18:   * Funktionen    : rmdir_flist, rmdir_dir,
                       rmdir_dlist
 19:   *
 20:   * Parameter     : rmdir_flist(path, flist);
 21:   *                 rmdir_dir(dir);
 22:   *                 rmdir_dlist(dlist);
 23:   *                 char            *path;
 24:   *                 FILE_DESC_LIST  flist;
 25:   *                 DIR_DESC_ITEM   dir;
 26:   *                 DIR_DESC_LIST   dlist;
 27:   *
 28:   * Aufgabe       :
 29:   *
 30:   * Löschen der in <dlist> befindlichen
        Verzeichnisse.
 31:   * Die Verzeichnisse müssen leer sein.
 32:   */
 33:
 34:  void rmdir_flist(path, list)
 35:  char            *path;
 36:  FILE_DESC_LIST list;
 37:  {   char  filepath[100];
 38:
 39:      while (list != NULL) {
 40:          if (list->fdesc.fattr & 0x10) {
 41:              sprintf(filepath, "%s\\%s", path,
 42:                      list->fdesc.fname);
 43:              if (Ddelete(filepath) < 0)
 44:                  fprintf(stderr,
 45:                          "rmdir: Can't delete
                                  %s\n",
 46:                          filepath);
 47:          }
 48:          list = list->next;
 49:      }
 50:  }
 51:
 52:  void rmdir_dir(dir)
 53:  DIR_DESC_ITEM dir;
 54:  {   if (strlen(dir.dirpath) == 3 &&
 55:          dir.dirpath[2] == '\\')
 56:          dir.dirpath[2] = 0;
 57:      rmdir_flist(dir.dirpath, dir.flist);
 58:  }
 59:
 60:  void rmdir_dlist(dlist)
 61:  DIR_DESC_LIST dlist;
 62:  {   if (dlist != NULL) {
 63:          rmdir_dir(*dlist);
 64:          rmdir_dlist(dlist->next);
 65:      }
 66:  }
 67:
 68:  /*
 69:   * Funktion      : rmdir
 70:   *
 71:   * Parameter     : ok = rmdir(argc, argv);
 72:   *                 BOOLEAN ok;
 73:   *                 short   argc;
 74:   *                 char    *argv[];
 75:   *
 76:   * Aufgabe       :
 77:   *
 78:   * Interpretierung der durch <argc> und <argv>
 79:   * spezifizierten Parameterliste gemäß den Fest-
 80:   * legungen des Kommandos rmdir.
 81:   */
 82:
 83:  BOOLEAN rmdir(argc, argv)
 84:  short argc;
 85:  char  *argv[];
 86:  {   DIR_DESC_LIST dlist;
 87:      BOOLEAN       start = TRUE;
 88:      short         i;
 89:
 90:      if (argc >= 2) {
 91:          for (i = 1; i < argc; i++) {
 92:              if (start) {
 93:                  dlist = built_dlist(argv[i],
 94:                                FALSE, TRUE,
 95:                                FALSE, 0);
 96:                  start = dlist->flist ==
 97:                          (FILE_DESC_LIST)ERROR;
 98:              }
 99:              else
100:                  dlist = expand_dlist(dlist,
                              argv[i],FALSE, TRUE,
102:                          FALSE, 0);
103:          }
104:          if (!start) {
105:              rmdir_dlist(dlist);
106:              drop_dlist(dlist);
107:          }
108:      }
109:      else {
110:          fprintf(stderr,
111:              "rmdir: need at least one
                      parameter\n");
112:          return(FALSE);
113:      }
114:      return(TRUE);
115:  }
116:
117:  void main(argc, argv)
118:  short argc;
119:  char  *argv[];
120:  {   if (!rmdir(argc, argv))
121:          exit(1);
122:      exit(0);
123:  }
```

```
 1:  /*
 2:   * Listing 1.17, Datei : mkdir.c
 3:   * Programm            : MKDIR - Erzeugen von
 4:   *                       Unterverzeichnissen
 5:   * Modifikationsdatum  : 15-Nov-89
 6:   * Abhängigkeiten      : stdio.h, osbind.h,
                              local.h
 7:   */
 8:
 9:  #include <stdio.h>
10:  #include <osbind.h>
11:  #include "local.h"
12:
13:  /*
14:   * Funktion     : mkdir
15:   *
16:   * Parameter    : ok = mkdir(argc, argv);
17:   *                BOOLEAN ok;
18:   *                short   argc;
19:   *                char    *argv[];
20:   *
21:   * Aufgabe      :
22:   *
23:   * Interpretierung der durch <argc> und <argv>
24:   * spezifizierten Parameterliste gemäß den Fest-
25:   * legungen des Kommandos mkdir.
26:   */
27:
28:  BOOLEAN mkdir(argc, argv)
29:  short argc;
30:  char  *argv[];
31:  {   short i;
32:
33:      if (argc >= 2) {
34:          for (i = 1; i < argc; i++) {
35:              if (Dcreate(argv[i]) != 0) {
36:                  fprintf(stderr,
37:                    "mkdir: can't create %s\n",
38:                    argv[i]);
39:                  return(FALSE);
40:              }
41:          }
42:      }
43:      else {
44:          fprintf(stderr,
45:             "mkdir: need at least one
                        parameter\n");
46:          return(FALSE);
47:      }
48:      return(TRUE);
49:  }
50:
51:  void main(argc, argv)
52:  short argc;
53:  char  *argv[];
54:  {   if (!mkdir(argc, argv))
55:          exit(1);
56:      exit(0);
57:  }
```

```
 1:  /*
 2:   * Listing 1.18, Datei : chmod.c
 3:   * Programm            : CHMOD - Ändern des
 4:   *                       Zugriffsmodus
 5:   * Modifikationsdatum  : 16-Nov-89
 6:   * Abhängigkeiten      : stdio.h, string.h,
                        osbind.h,
 7:   *                       local.h, atom.h, expand.h
 8:   */
 9:
10:  #include <stdio.h>
11:  #include <string.h>
12:  #include <osbind.h>
13:  #include "local.h"
14:  #include "atom.h"
15:  #include "expand.h"
16:
17:  /*
18:   * Funktionen   : chmod_flist, chmod_dir,
                      chmod_dlist
19:   *
20:   * Parameter    : chmod_flist(path, flist, setrw);
21:   *                chmod_dir(dir, setrw);
22:   *                chmod_dlist(dlist, setrw);
23:   *                 char            *path;
24:   *                 FILE_DESC_LIST  flist;
25:   *                 BOOLEAN         setrw;
26:   *                 DIR_DESC_ITEM   dir;
27:   *                 DIR_DESC_LIST   dlist;
28:   *
29:   * Aufgabe      :
30:   *
31:   * Ändern des Zugriffsmodus für alle innerhalb von
32:   * <dlist> befindlichen Dateien. Je nach Wert von
33:   * <setrw> wird schreibender Zugriff entweder ver-
34:   * geben (<setrw> ist TRUE) oder entzogen (<setrw>
35:   * ist FALSE).
36:   */
37:
38:  void chmod_flist(path, list, setrw)
39:  char           *path;
40:  FILE_DESC_LIST list;
41:  BOOLEAN        setrw;
42:  {   char  filepath[100];
43:      short attr;
44:
45:      while (list != NULL) {
46:          if (!(list->fdesc.fattr & 0x10)) {
47:              sprintf(filepath, "%s\\%s", path,
48:                  list->fdesc.fname);
49:              if (setrw)
50:                  attr = list->fdesc.fattr | 0x01;
51:              else
52:                  attr = list->fdesc.fattr & 0xfe;
53:              Fattrib(filepath, 1, attr);
54:          }
55:          list = list->next;
56:      }
57:  }
58:
59:  void chmod_dir(dir, setrw)
60:  DIR_DESC_ITEM dir;
61:  BOOLEAN       setrw;
62:  {   if (strlen(dir.dirpath) == 3 &&
63:          dir.dirpath[2] == '\\')
64:          dir.dirpath[2] = 0;
65:      chmod_flist(dir.dirpath, dir.flist, setrw);
66:  }
67:
68:  void chmod_dlist(dlist, setrw)
69:  DIR_DESC_LIST dlist;
70:  BOOLEAN       setrw;
71:  {   if (dlist != NULL) {
72:          chmod_dir(*dlist, setrw);
73:          chmod_dlist(dlist->next, setrw);
74:      }
75:  }
76:
77:  /*
78:   * Funktion     : chmod
79:   *
80:   * Parameter    : ok = chmod(argc, argv);
81:   *                BOOLEAN ok;
82:   *                short   argc;
83:   *                char    *argv[];
84:   *
85:   * Aufgabe      :
86:   *
87:   * Interpretierung der durch <argc> und <argv>
88:   * spezifizierten Parameterliste gemäß den Fest-
89:   * legungen des Kommandos chmod.
90:   */
91:
92:  BOOLEAN chmod(argc, argv)
93:  short argc;
94:  char  *argv[];
95:  {   DIR_DESC_LIST dlist;
96:      BOOLEAN       setrw,
97:                    start = TRUE;
98:      short         i;
99:
100:     if (argc >= 3) {
101:         convupper(argv[1]);
102:         if (strcmp(argv[1], "+W") == 0 ||
103:             strcmp(argv[1], "-W") == 0) {
104:             setrw = strcmp(argv[1], "-W") == 0;
105:             for (i = 2; i < argc; i++) {
106:                 if (start) {
107:                     dlist = built_dlist(argv[i],
108:                                         FALSE,
109:                                         TRUE,
```

→

```
110:                                     FALSE, 0);
111:                    start = dlist->flist ==
112:                            (FILE_DESC_LIST)ERROR;
113:                }
114:                else
115:                    dlist = expand_dlist(dlist,
116:                                         argv[i],
117:                                         FALSE,
118:                                         TRUE,
119:                                         FALSE,
                                           0);
120:            }
121:            if (!start) {
122:                chmod_dlist(dlist, setrw);
123:                drop_dlist(dlist);
124:            }
125:        }
126:        else {
127:            fprintf(stderr, "chmod: SYNOPSIS:
                                 %s\n",
128:                "chmod [ +w ] [ -w ]
               Filename...");
129:            return(FALSE);
130:        }
131:    }
132:    else {
133:        fprintf(stderr,
134:            "chmod: need at least two
             parameter\n");
135:        return(FALSE);
136:    }
137:    return(TRUE);
138: }
139:
140: void main(argc, argv)
141: short argc;
142: char  *argv[];
143: {   if (!chmod(argc, argv))
144:         exit(1);
145:     exit(0);
146: }
```

```
  1: /*
  2:  * Listing 1.19, Datei : touch.c
  3:  * Programm             : TOUCH - Aktualisierung
                              der
  4:  *                        Modifikationszeit von
  5:  *                        Dateien
  6:  * Modifikationsdatum   : 16-Nov-89
  7:  * Abhängigkeiten       : stdio.h, string.h,
                              osbind.h,
  8:  *                        local.h, atom.h,
                              expand.h
  9:  */
 10:
 11: #include <stdio.h>
 12: #include <string.h>
 13: #include <osbind.h>
 14: #include "local.h"
 15: #include "atom.h"
 16: #include "expand.h"
 17:
 18: /*
 19:  * Funktionen    : touch_flist, touch_dir,
                       touch_dlist
 20:  *
 21:  * Parameter     : touch_flist(path, flist);
 22:  *                 touch_dir(dir);
 23:  *                 touch_dlist(dlist);
 24:  *                 char           *path;
 25:  *                 FILE_DESC_LIST flist;
 26:  *                 DIR_DESC_ITEM  dir;
 27:  *                 DIR_DESC_LIST  dlist;
 28:  *
 29:  * Aufgabe       :
 30:  *
 31:  * Aktualisierung der Modifikationszeit aller in
 32:  * <dlist> befindlichen Dateien oder
     Verzeichnisse.
 33:  */
 34:
 35: void touch_flist(path, list)
 36: char            *path;
 37: FILE_DESC_LIST list;
 38: {   char  filepath[100];
 39:     short fhandle;
 40:
 41:     while (list != NULL) {
 42:         if (!(list->fdesc.fattr & 0x10)) {
 43:             sprintf(filepath, "%s\\%s", path,
 44:                     list->fdesc.fname);
 45:             if (!ftouch(filepath))
 46:                 fprintf(stderr,
 47:                         "touch: Can't touch %s",
 48:                         filepath);
 49:         }
 50:         list = list->next;
 51:     }
 52: }
 53:
 54: void touch_dir(dir)
 55: DIR_DESC_ITEM dir;
 56: {   if (strlen(dir.dirpath) == 3 &&
 57:         dir.dirpath[2] == '\\')
 58:         dir.dirpath[2] = 0;
 59:     touch_flist(dir.dirpath, dir.flist);
 60: }
 61:
 62: void touch_dlist(dlist)
 63: DIR_DESC_LIST dlist;
 64: {   if (dlist != NULL) {
 65:         touch_dir(*dlist);
 66:         touch_dlist(dlist->next);
 67:     }
 68: }
 69:
 70: /*
 71:  * Funktion      : touch
 72:  *
 73:  * Parameter     : ok = touch(argc, argv);
 74:  *                 BOOLEAN ok;
 75:  *                 short   argc;
 76:  *                 char    *argv[];
 77:  *
 78:  * Aufgabe       :
 79:  *
 80:  * Interpretierung der durch <argc> und <argv>
 81:  * spezifizierten Parameterliste gemäß den Fest-
 82:  * legungen des Kommandos touch.
 83:  */
 84:
 85: BOOLEAN touch(argc, argv)
 86: short argc;
 87: char  *argv[];
 88: {   DIR_DESC_LIST dlist;
 89:     BOOLEAN       start = TRUE;
 90:     short         i;
 91:
 92:     if (argc >= 2) {
 93:         for (i = 1; i < argc; i++) {
 94:             if (start) {
 95:                 dlist = built_dlist(argv[i],
 96:                                     FALSE, TRUE,
 97:                                     FALSE, 0);
 98:                 start = dlist->flist ==
 99:                         (FILE_DESC_LIST)ERROR;
100:             }
101:             else
102:                 dlist = expand_dlist(dlist,
                                         argv[i],
103:                                      FALSE, TRUE,
104:                                      FALSE, 0);
105:         }
106:         if (!start) {
107:             touch_dlist(dlist);
108:             drop_dlist(dlist);
109:         }
110:     }
111:     else {
112:         fprintf(stderr,
113:             "touch: need at least one
                        parameter\n");
114:         return(FALSE);
115:     }
116:     return(TRUE);
117: }
118:
119: void main(argc, argv)
120: short argc;
121: char  *argv[];
122: {   if (!touch(argc, argv))
123:         exit(1);
124:     exit(0);
125: }
```



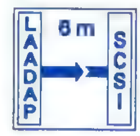
KLAPPE!

Die Erfolgsserie geht weiter!

Subsysteme für ATARI-ST-/TT-, AT- und MAC-Rechner sowie Sampler-Systeme

Festplatten von 32 MB bis 2 GB • Wechselplatten mit 44 MB • Bandlaufwerke mit 155 MB • Optische Laufwerke mit 2x 320 MB

Durch modernste SCSI-Technik und den neuentwickelten Hostadapter "LAADAP" setzen unsere Syteme neue Maßstäbe!



Am 25-pol. SCSI-Port können weitere SCSI-Geräte über eine Leitungslänge von bis zu 6 m angeschlossen werden (Insgesamt 8 SCSI-Geräte sind möglich).



Der 25-pol. SCSI-Port kann auch als Eingang benutzt werden, sodaß das System ohne Umschaltung auch an Computersysteme mit SCSI-Port angeschlossen werden kann (Diverse Sampler, AT-Rechner, Apple-Mac).



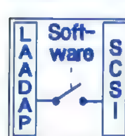
Gleichzeitiger Anschluß über DMA- und SCSI-Port möglich. Dadurch kann abwechselnd vom Atari- und SCSI-Gerät auf das gleiche SD-System zugegriffen werden (besonders interessant bei der Wechslplatte SD44E und dem Streamer SDT155).



Zwei ATARI -ST/-TT-Rechner mit jeweils einem SD-System können über ein 6m langes SCSI-Kabel verbunden werden. Jeder der beiden Atari-Rechner kann auf die Daten seiner Festplatte und die des anderen Rechners zugreifen.



Die Umschaltung der SCSI-Ports ist vollautomatisch.



Der 25-pol. SCSI-Port kann per Software gesperrt werden, sodaß die zusammengeschalteten Systeme völlig unabhängig voneinander arbeiten können.



Das Softwarepaket SDSOFT (ATARI ST/TT) ist im Lieferumfang enthalten, incl. dem neuartigen LFSS (LACOM-FAT-SICHERUNGS-SYSTEM) und dem Harddisk-Analyser/-Optimizer SDCHECK.



Ein Hardware-Schreibschutz, der per Software betätigt wird, schützt vor Schreibzugriffen (Virus)! Ist dieser Schreibschutz aktiv, so kann nur noch über den SCSI-Port schreibend auf die Platte zugegriffen werden.



Eine echte PARITY-OPTION, die nicht nur bei einem Schreibvorgang ein Prüfbit erzeugt, sondern dieses auch bei Lesevorgägen auswertet, erkennt leitungsbedingte Übertragungsfehler. Der LACOM SD-DRIVER für ATARI ST/-TT führt dann autom. eine Wiederholung durch.



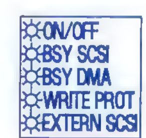
Eine eingebaute Einschaltverzögerung wartet bei einem Kaltstart eine einstellbare Zeit lang und läßt erst dann den ATARI-ST booten. Dadurch werden abgebrochene Systemstarts durch noch nicht hochgelaufene Festplatten vermieden.



Eine Thermoregelung sorgt dafür, daß der eingebaute Lüfter nur bei Erwärmung betrieben wird. Der Lüfter wird dann entsprechend der Temperatur geregelt. Selbst im höheren Drehzahlbereich ist der Lüfter fast unhörbar!



Mittels Software können interne und externe SCSI-Geräte zurückgesetzt werden (Hardware-RESET).



Das INFO-PANEL zeigt die wichtigsten Zustände Ihres SD-Systems über fünf LED's an:
ON/OFF • SCSI-BUSY • DMA-BUSY
Write-Protect • SCSI-Extern



Ältere SD-Systeme mit eingebautem "LAADAP" können problemlos aufgerüstet werden.

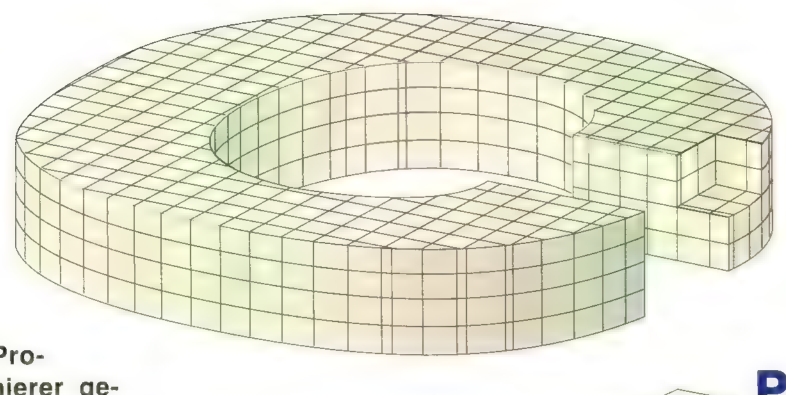
Weitere Standard-Eigenschaften: Gepufferter DMA-OUT • Laufwerke auf Stoßdämpfer befestigt • Aufrüstbar mit 5¼" Laufwerk • Alle SCSI-Gruppencodes • SCSI-Adresse von außen einstellbar • komplett anschlußfertig u.v.m.

Fordern Sie unser kostenloses Informationsmaterial noch heute an!
Unsere Systeme erhalten Sie bei Ihrem Fachhändler und in allen SCHAULANDT-Filialen.

LACOM Ilias Lazaridis

Stöckmannstr. 56 • 4200 Oberhausen 1 • Tel. 02 08 / 80 50 46 • Fax 02 08 / 2 52 74

Händleranfragen sind uns stets willkommen - Wir suchen noch Generaldistributoren für: Schweiz • Österreich • Niederlande



5 Programmiertechnik & OOP

Während Programmierer gemeiner Sprachen begeistert mit 100 Sachen durch die Platine heizen, schaut ein C-Freak erst auf, wenn die Elektronenschallmauer durchbrochen ist. Doch der leistungsstarke Motor C braucht zuverlässige und geübte Rennfahrer, um nicht aus der Bahn zu geraten. Heute wollen wir uns anschauen, welche Spritmischung so ein Motor braucht, wie man auch die härtesten Strecken bewältigt und welche Vorteile die objektorientierte Fahrweise (OOF) bringt.

Zurück auf dem harten Boden der Realität, werden Sie sicher schon von Leuten gehört haben, die immer noch glauben, daß C nicht den Ruhm verdient, den es geerntet hat. Die Argumente dieser hartnäckigen und durchaus bewundernswerten Personen beschränken sich oft auf die angeblich kryptologische Schreibweise und hohe Fehlerhäufigkeit von C. Im ersten Teil des Artikels werde ich zeigen, mit welchen einfachen Mitteln man jede zweitrangige Programmiersprache (ich spare mir hier die Aufzählung) in puncto Übersichtlichkeit und Zuverlässigkeit um Längen schlagen kann. Im zweiten Teil erhalten Sie einen kurzen Überblick über objektorientierte Erweiterungen.

Die Form

Wer sich schon öfter mal PD-Listings angeschaut hat, der weiß, wie unterschiedlich diese aussehen können, sowohl was den logischen Aufbau als auch einzelne Textzeilen angeht. Je nach dem Willen des Programmierers entstehen dadurch mehr oder weniger lesbare Listings. Mit den sehr frühen Versionen von Turbo-C wurde ein hübsches, lauffähiges Programm geliefert, das wie eine Lokomotive aussah. Es gibt einige Grundsätze, an die man sich als C-Programmierer halten sollte, um gut lesbare Programme zu schreiben. Diese zeichnen sich dadurch aus, daß man sich jederzeit wieder schnell zurechtfindet, wenn man das Listing in die Hand nimmt. Auch sollten sich Betrachter, die das Programm nicht geschrieben haben, gleich wohlfühlen.

Programme sollten ab einer gewissen Größe aus mehreren kleineren Modulen bestehen, die nicht die 50kB-Grenze, das sind ungefähr 2000 Zeilen, überschreiten. Diese Module werden dann mit dem Linker zu einem Programm gebunden. Der Linker betrachtet ein Modul als ein Stück Code, das einige Variablen und Routinen enthält, die von außen angesprochen werden, und das selbst auf andere zugreift. Alle diese sogenannten externen Referenzen zusammen bilden die Schnittstelle eines Moduls. Die Schnittstelle sollte innerhalb eines Listings hervorgehoben werden. Wir unterscheiden zwischen Deklarationen und Definitionen. Deklarationen, bei Funktionen auch Prototypen, beschreiben eine Variable oder Routine nur, während Definitionen diese tatsächlich definieren. Erstere können dazu verwendet werden, dem Compiler externe Elemente bekanntzumachen. Es entstehen also folgende Möglichkeiten: Eine Funktion/Variable kann lokal zu einem Modul, global, also von innen und außen zugreifbar, oder extern sein. Jede Möglichkeit gehört in einen eigenen Abschnitt im Listing.

Die logische Unterteilung in Abschnitte muß auch für die restlichen Elemente von C eingehalten werden. So gehören zum Beispiel *defines*, *typedef* und *enums* alle in einen eigenen Abschnitt. Listing 1 zeigt, wie diese Unterteilung in der Praxis aussehen kann.

Auch einzelne Zeilen innerhalb eines Listings müssen gut aufgebaut sein. Niemals darf eine Zeile länger als 80 Zeichen sein! Das würde einem Betrachter mit normalem Monitor Schwierigkeiten bereiten, denn er müßte am Zeilenende noch ein Stück nach rechts scrollen. Auch hat es sich eingebürgert, daß eine Zeile nur einen einzigen Befehl enthält.

Groß- und Kleinschreibung verwendet man im täglichen Leben, um einzelne Wortarten auf den ersten Blick voneinander unterscheiden zu können. Dasselbe Prinzip hat sich auch in C durchgesetzt. So werden Schlüsselwörter immer klein geschrieben! Namen für Variablen oder Funktionen sollten in C keine Großbuchstaben enthalten und nach Möglichkeit englisch sein. Sie sollten aber nicht überlange Namen verwenden. Zur besseren Unterscheidung davon, werden Ersatz-Strings (#define) und zusammengesetzte Typen (struct, union) immer groß geschrieben.

```
#define DEBUG 0
typedef struct
{
   char    *name;
   char    *phone;
} MEMBER;
```

Eine beliebte Technik, um Programme portabel zu halten, ist die Verwendung von Ersatztypen. Beispiel:

```
#define WORD short
```

Das ist zwar empfehlenswert, doch sollten Sie lieber die folgende Zeile verwenden:

```
typedef word short;
```

Hier kann nämlich ein einfacher Typ von einem zusammengesetzten unterschieden werden.

Leerstellen

Mit der selben Sorgfalt, mit der Leerstellen in Texten verwendet werden, um eine übersichtliche Darstellung zu ermöglichen, müssen diese auch in den Listings gesetzt werden. C stellt eine Menge von Operatoren zur Verfügung. Man unterscheidet zwischen den unären und den binären Operatoren, je nachdem, ob sie auf einen oder zwei Operanden wirken. Jeder Operator besitzt ein eigenes Verhalten, was die Leerstellen angeht.

Bei folgenden Operatoren befinden sich Leerstellen an beiden Seiten: arithmetische (*, +, %, ...) und logische (&&, ||) Operatoren, Relationen (==, <=, ...), Zuweisungen (=, +=, -=, ...) und Bit-Manipulationen (&, |, ^ ohne ~). Beispiele:

```
a += b & c;
if(a < b)
   b = a;
```

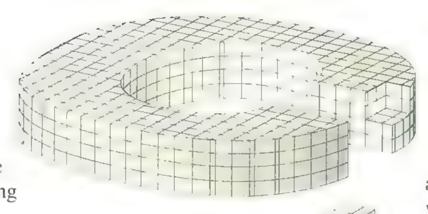
Die restlichen Operatoren, die noch verbleiben, erfordern keine Leerzeichen:

```
memberp->name = member[1].name;
if(!a)
   func();
```

Verbleiben noch das Komma, der Doppelpunkt und das Semikolon. Sie werden wie in normalen Texten verwendet, indem vor dem Zeichen keine, dafür aber danach eine Leerstelle steht:

```
for(i = 0; i < 100; i++)
   add(i, 100);
```

Die Klammern nach Befehlen oder Funktionen sind nicht durch Leerzeichen von dem Namen getrennt. C-Compiler erlauben nach *sizeof* und *return* die Möglichkeit, die Klammern wegzulassen. Sie sollten diese Möglichkeit nicht ausnutzen und immer Klammern verwenden!



5

Einrückungen

Kontrollkonstrukte, wie zum Beispiel *for*, *switch* oder *if*, die in strukturierten Sprachen Verwendung finden, treten besonders gut hervor, wenn man Befehle innerhalb dieser einrückt. Nach einem eingerückten Ausdruck folgt immer eine Leerzeile.

```
for( ; ; )
   foobar();
next();
```

Sollten Blöcke von Befehlen angewendet werden, gilt grundsätzlich: die öffnende geschweifte Klammer befindet sich in der nächsten Zeile, gleich weit eingerückt wie der Befehl vor ihr. Erst die tatsächlichen Befehle werden einen Tabulator weiter gesetzt. Die schließende Klammer gehört wieder in eine neue Zeile, mit derselben Einrückung, die die öffnende besitzt. Nach einem Block folgt keine Leerzeile.

```
if(test == TRUE)
{
   do_it();
   ...
}
next();
```

Die Verschachtelung der Befehle im Listing zu verdeutlichen, hat sich als sehr nützlich erwiesen. Probleme treten aber auf, wenn viele schließende Klammern vorkommen oder eingerückte Blöcke die Länge eines Bildschirms überschreiten. Man kann dann nicht mehr erkennen, welche Klammer zu welchem Befehl gehört. Wir können in C die Lösung von ADA verwenden, indem wir nach der Klammer in einem Kommentar den passenden Befehl angeben:

```
switch
{
   ..
   ...
} /* switch */
```

Diese Kommentare empfehlen sich erst ab einer Blocklänge von 20 Zeilen. Einrükkungen können Sie übrigens auch mit den Präprozessor-Direktiven verwenden:

```
#ifdef DEBUG
#        undef DEBUG
#endif
```

Ich habe oben einige der wichtigsten Prinzipien aufgezählt. Sie sollten sich so weit wie möglich an diese annähern, um das einheitliche Aussehen von C-Programmen zu gewährleisten. Auch wenn Sie sich in einzelnen Punkten nicht an die Ausführungen halten, sollten Sie wenigstens darauf achten, daß Sie Ihren Stil konsequent durchziehen.

Der Inhalt

Portabilität, Hierarchie und Modularität sind die drei wichtigsten Stichwörter, wenn Sie gute Programme schreiben wollen. Portabilität ist gerade jetzt hochaktuell. Seit der TT erhältlich ist, zeigt sich nämlich, wer sauber programmiert hat und wer nicht. Die höchste Portabilität erreichen Sie natürlich, indem Sie keine Annahmen über das Betriebssystem und die Hardware des Zielsystems machen. Alle Funktionen die Sie dann benutzen dürfen, finden Sie in der Standardbibliothek, die von ANSI definiert wurde und auf jeder Umgebung vorhanden sein muß. Sie haben aber sicher schon bemerkt, daß in dieser Bibliothek keine Routinen enthalten sind, die z.B. Grafik oder Sound unterstützen. Ohne diese ist die Erstellung benutzerfreundlicher Programme aber nicht möglich. Auf dem ST hat man glücklicherweise das GEM bereits ins Betriebssystem integriert, so daß Sie in der Lage sind, indem Sie GEM verwenden, Programme zu schreiben, die auf allen Atari Computern lauffähig sind. Durch den Einsatz von GEM auf den IBM-Kompatiblen ist sogar die Portierung zwischen diesen grundverschiedenen Computern möglich. Sollten aber Portierungen auf UNIX erwünscht sein, gibt es nur zwei Wege: entweder Sie verwenden überhaupt keine Benutzeroberflächen, oder Sie schreiben die Programme so portabel, daß eine Portierung einfach ist. Das ist aber nur über große Umwege möglich. Die naheliegende Lösung, eine eigene Benutzeroberfläche zu schreiben, ist auf dem ST nicht erlaubt!

Wenn man nun aber gewisse Annahmen über das System machen muß, sollten die Routinen, die sich auf diese Annahmen stützen, sorgfältig abgegrenzt und ausgesondert werden. Sie gehören in eigene Module! Muß das Programm portiert werden, kann man, ohne größere Änderungen am eigentlichen Programm, diese Module durch neue ersetzen.

Auch beim Handling mit der Hardware und dem Betriebssystem muß man einige Regeln einhalten, um mit neuen Versionen keine Probleme zu bekommen. Hier gilt grundsätzlich: Halten Sie sich immer an die Dokumentation! Verwenden Sie Bücher wie [1] und [2], um Unklarheiten

zu beseitigen. Benutzen Sie nie undokumentierte Eigenschaften! Die von einigen Programmierern heißgeliebten Line-A-Routinen sind in neuen Systemen zwar weiterhin vorhanden, sie unterstützen aber nicht deren besondere Eigenschaften, d.h. Sie können vergessen, daß es sie jemals gab.

Leider konnten Fehler bei der Erstellung des TOS nicht vermieden werden. Man muß also wissen, wie man mit diesen Fehlern umzugehen hat, um nicht bei der nächsten Revision des Betriebssystems durch die Behebung der Fehlerstelle unkorrekte Ergebnisse zu bekommen. Meistens entdecken Journalisten und Programmierer solche Unzulänglichkeiten und geben diese dem Leser bekannt. Das beste wäre nun, im Programm selber überhaupt keine Rücksichten auf diese Fehler zu nehmen und dann einen Patch zu schreiben, der sie behebt. Eine andere Methode bietet sich besonders bei falscher Beschriftung der Parameter an. Man verwendet wie oben die eigentlich korrekten Parameter und wandelt sie über den Präprozessor in die momentan gültigen um.

Beispiel:

```
evnt_button(a, b, c, x, y, s, k);
(eigentlich richtig)
evnt_button(0x100 + a, 0x3, 0, x,
y, s, k); (GEM < V2.0)
```

Lösung:

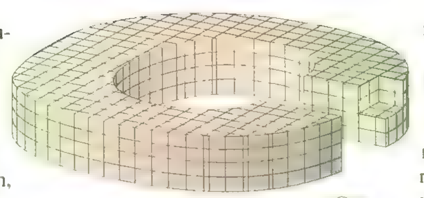
```
#define evnt_button(a, b, c, x, y,
s, k) \
evnt_button(0x100 + a, 0x3, 0, x,
y, s, k)
```

Ein nach der Dokumentation korrekter Ausdruck wird durch dieses *#define* in die falsche, dafür aber lauffähige Version gewandelt. Sollte nun dieser Fehler mal behoben werden, genügt eine Rekompilation ohne dieses *#define*. Leider liefert Atari die neuen Versionen nicht kostenlos oder gegen alte ROMs aus. Sie müssen, wenn Sie neue Features benutzen, immer zuerst abfragen, ob sich diese denn überhaupt im System befinden. Listing 2 stellt Ihnen eine Routine zur Verfügung, mit der Sie feststellen können, welches Betriebssystem sich in dem Rechner befindet.

Damit wäre der Teil zu den Richtlinien abgeschlossen. Ich habe versucht, die wichtigsten Elemente aufzuzählen, und wenn Sie sich daran halten, dürften Sie keine Schwierigkeiten in dieser Richtung bekommen.

OOP

Seit einiger Zeit kann man in allen wichtigen Computerzeitschriften ein Phänomen beobachten: Die sogenannte objektorientierte Programmierung (OOP) verursachte einiges Erstaunen bei den Programmierern. Smalltalk, der Vater aller objektorientierten Sprachen, löste eine Art Lawine aus. Viele neue Programmiersprachen wurden entwickelt, die diese neue Programmierweise ermöglichen. Liebhaber *alter* Sprachen wie Pascal oder C mußten natürlich mithalten, und so entstanden die ersten Hybridsprachen, die die guten alten Eigenschaften erhielten und gleichzeitig objektorientierte Hypes mit in die Sprache einbezogen. C entwickelte sich hauptsächlich in zwei Strömungen: C++ und Objective-C. Beide haben ihre Vor- und Nachteile: C++ will eine konsequente Weiterentwicklung von C sein, die sich nicht nur auf OOP beschränkt, sondern auch noch bekannte Ineffizienzen von C beheben will. Objective-C erweitert C dagegen nur um smalltalk-ähnliche, objektorientierte Features.



Doch was ist OOP eigentlich? Die genaue Definition fehlt bis heute! Darunter versteht man aber eine Programmierweise, die mit Objekten arbeitet. Ein Objekt ist ein Sprachkonstrukt, das in sich abgeschlossen ist. Es kann Nachrichten empfangen und versenden. Wenn man in strukturierten Sprachen einer Routine sagt, was sie mit welcher Struktur machen soll, dann versendet man in OOP eine Nachricht an das betreffende Objekt. Programmiertechnisch ist ein Objekt beispielsweise eine Vereinigung einer C-Struktur mit den Funktionen, die mit dieser Struktur operieren. Hat man erst einmal ein Objekt definiert, kann man seine Eigenschaften an neue Objekte vererben. Diese können neue Eigenschaften definieren, alte überlagern oder gar löschen. Das Programm ist dann eine Hierarchie von Objekten. Die Funktionen innerhalb eines Objekts, Methoden genannt, werden nicht schon beim Linken mit ihren Aufrufern verbunden, sondern erst während der Laufzeit. Das hat den Vorteil, daß Objekte während der Laufzeit vertauschbar sind. Durch die Wiederverwendung von Objekten verhindert man, daß Räder neu erfunden werden.

C++

C++ wurde von Bjarne Stroustrup entwickelt und ist aus historischen Gründen nicht nur ein objektorientierter Zusatz, sondern vielmehr eine komplette Weiterentwicklung von C. Stroustrup hat versucht, alle Erweiterungen im Stil von C zu halten, was ihm meiner Meinung nach nicht gelungen ist. Er hat gegen einen der wichtigsten Grundsätze von C verstoßen, der besagt, daß sich die Sprache selbst nicht mit Ressourcen beschäftigt (um diese Aufgabe kümmert sich die Bibliothek), sondern nur Kontrollkonstrukte bietet. Wie in Pascal, gibt es nämlich auch in C++ die Befehle (eigentlich Operatoren) *new* und *delete* zur Verwaltung des Speichers. C++ unterstützt alle oben aufgeführten OOP-Features und beherrscht auch multiple Vererbung, d.h., daß ein Objekt auch die Eigenschaften mehrerer Objekte erben kann. Das Überlagern von Operatoren wird besonders die Mathematiker unter Ihnen freuen. Man kann einen beliebigen Operator einfach mit einer anderen Bedeutung überlagern und somit auch mit Strukturen übersichtlich rechnen. Zum Beispiel kann man mit komplexen Zahlen rechnen, indem man, wie in der Mathematik, die arithmetischen Operatoren benutzt. Ein Operator verhält sich dann wie eine Funktion und muß auch so definiert werden. Allen, die sich besonders für C++ interessieren, empfehle ich [3], vor allem aber [4].

Man sollte nicht verschweigen, daß die aktuelle Entwicklung in Richtung C++ geht, und so hat man sich entschieden, C++ zu standardisieren. ANSI C++ Komitee (X3J16) hat diese Aufgabe bekommen und schickt sich gerade an, Informationen über die Erfahrung mit dieser Sprache zu sammeln.

Wir werden sehen, ob Borland sich entscheiden wird, C++ auch für den Atari auszuliefern, oder ob wir uns mit der GNU-Version zufriedengeben müssen. Das letzte Wort ist hier noch nicht gesprochen, und wir werden Sie auf dem laufenden halten.

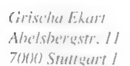
Objective-C

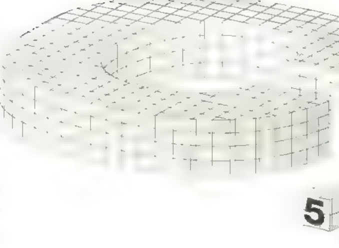
Objective-C hat für Atari-Anwender einen entscheidenden Vorteil gegenüber C++: es ist eine lauffähige Version für den ST vorhanden [5]. Mit Objective-C läßt sich schon jetzt fast problemlos arbeiten, und ich verweise alle, die mehr über die Entwicklung in dieser Programmiersprache wissen wollen, auf eine der nächsten Ausgaben der ST-Computer.

Nachwort

So, das war's mal wieder. In der nächsten Folge dieser Artikelreihe beschäftigen wir uns mit ... Aber leider (oder auch nicht?!) gibt es keine nächste Folge mehr. Tja, es

ist soweit: Wir sind am Ende angekommen, bevor es eigentlich wirklich losging. Ich habe versucht, einige der Themen aufzugreifen, die meiner Meinung nach in der üblichen Literatur unterbewertet wurden. Wenn Sie meinen, daß ich was Wichtiges vergessen hätte, oder wenn Sie Ihre Kritik loswerden wollen, dann schreiben Sie mir doch an die untenstehende Adresse. Stay with techno!


Grischa Ekart
Abelsbergstr. 11
7000 Stuttgart 1



Literatur:

[1] Jankowski, Reschke, Rabich: Atari ST Profibuch, 1987, Sybex-Verlag

[2] D. Geiß, J. Geiß: Vom Anfänger zum GEM-Profi 1990, Hüthig ISBN 3-7785-1792-9

[3] B. Stroustrup: Die C++ Programmiersprache, 1987, Addison Wesley ISBN 3-92511872-1

[4] S. Lippman: C++ Einführung und Leitfaden, 1990, Addison Wesley ISBN 3-89319-276-6

[5] G. Ekart: M:OOP Magic OOP?, ST-Computer 9/1990

```
 1:  /**************************************************/
 2:  /*              TOS_VER.C  von 14.11.1990         */
 3:  /*          (w) 1990 by Grischa Ekart             */
 4:  /* erstellt mit Turbo C V1.0 von BORLAND GmbH     */
 5:  /**************************************************/
 6:  #include <tos.h>
 7:
 8:  /* function prototypes */
 9:
10:  int   tos_version(void);
11:
12:  /* function definitions */
13:
14:  int
15:  tos_version(void)
16:  {
17:     void  *ssp;
18:     int   version;
19:
20:     ssp = (void *)Super(0L);
21:     version = *(int *)(*((long *)0x4f2) + 2);
22:     Super(ssp);
23:     return(version);
24:  }
```

Listing 2

```
 1:  /**************************************************/
 2:  /*              HYPE.C  von 14.11.1990            */
 3:  /*     (g) & (p) 1990 by Manuella Bartolome       */
 4:  /**************************************************/
 5:
 6:  /*  Externe Variablen  */
 7:
 8:  extern  int errno;
 9:  ...
10:
11:  /*  Globale Variablen  */
12:
13:  long     count;
14:  ...
15:
16:  /*  Lokale Variablen   */
17:
18:  static  char *name;
19:  ...
20:
21:  /*  Externe Funktionen  */
22:
23:  void     test(void);
24:  ...
25:
26:  /*  Globale Funktionen  */
27:
28:  int      main(int argc, char **argv);
29:  ...
30:
31:  /*  Lokale Funktionen  */
32:
33:  static  void quit(int termcode);
34:  ...
```

Listing 1

ROCKUS



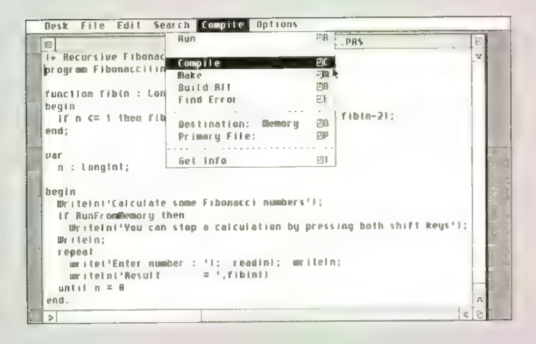




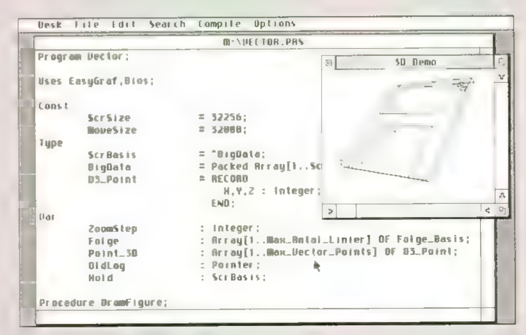
Das integrierte PASCAL-Entwicklungssystem für den ATARI ST/TT

Mit seinen Leistungsdaten und Funktionen definiert MAXON PASCAL neue Maßstäbe in der Software-Entwicklung. Herausragender Komfort, höchste Leistungsfähigkeit und Effizienz sowie größtmögliche Kompatibilität zu verschiedenen bestehenden Standards machen dieses integrierte Entwicklungssystem zu einem Meilenstein in der Software-Entwicklung.

- Der komfortable Multi Window Editor bietet unter anderem folgende Vorteile:
  Compiler und Linker werden vom Editor aus aufgerufen und benötigen keine Ladezeiten.
  Sämtliche Programme können wahlweise komplett im Speicher oder auch über Diskette bearbeitet werden.
  Interaktive Fehlererkennung (Compiler- und Programmfehler).
- Der extrem kompakte und schnelle Single Pass-Compiler belegt wenig Speicherplatz. Compilierungsgeschwindigkeit: mehr als 20.000 Zeilen/min. auf einem normalen ST.



- Extrem hohe Ausführungsgeschwindigkeit der fertigen Programme durch effektive Code-Optimierung.
- MAXON PASCAL arbeitet mit UNITs, wodurch extrem schnelle Durchlaufzeiten des Compilers realisiert werden können.



- Das UNIT-Prinzip ermöglicht den einfachen modularen Aufbau von Programmen. Oft benötigte UNITs können zu Libraries zusammengefaßt und resident im Speicher gehalten werden, wodurch beim Compilieren extreme Zeitvorteile entstehen.
- Integrierter Inline-Assembler
- Weitgehende Kompatibilität zu Turbo Pascal 5.0 auf dem PC.
- Die Übernahme von ST-Pascal-Programmen wird durch eine spezielle UNIT unterstützt.
- Schnelle, IEEE-kompatible Floating-Point-Library für Fließkommaarithmetik (Single, Double, Ext. Precision).
- Mitgeliefert wird eine Command-Line-Version, die das Compilieren über Batch-Dateien ermöglicht, sowie eine 68881-kompatible Floating-Point-Version.
- Eine komplexe und funktionsabhängige Hilfefunktion unterstützt den Anwender in Fragen auf die Pascal-Syntax, den Editor und die generelle Bedienung.

MAXON PASCAL ist zum unverbindlich empfohlenen Verkaufspreis von DM 259,- erhältlich.
Gerne nennen wir Ihnen einen Händler in Ihrer Nähe oder senden Ihnen weitere Informationen zu.



MAXON Computer GmbH • Schwalbacher Straße 52 • D-6236 Eschborn
Tel. 06196/481811 • Fax 06196/41885

MODULA

MARZIPAN

Sweet Secrets Nº 3

In dieser dritten Folge über die Feinheiten von Modula-2 geht es um RECORDs, insbesondere in ihrer speziellen Form als variante RECORDs.

Ein RECORD - in der Informatik auch Verbund genannt - ist eigentlich nichts weiter als eine Methode, verschiedene Variablen zusammenzufassen. Logisch zusammengehörige Daten werden programmtechnisch zusammengehörig notiert und können dann auch auf einen Schlag gemeinsam beispielweise für eine Zuweisung benannt werden.

Der einfache Aufbau eines RECORDs sollte Ihnen bekannt sein, dennoch ein kleines Beispiel:

```
VAR koordinate,o  :  RECORD
                       x,y  :  INTEGER
                     END;
```

Wir haben hier zwei Variablen, die eine Koordinate auf einem Zeichengerät - beispielsweise dem Bildschirm - enthalten sollen. Da wir mit zwei Dimensionen arbeiten, haben sie einen x- und einen y-Bestandteil. Diese Werte werden in den INTEGER-Variablen *x* und *y* abgelegt. Logisch gehören sie eng zusammen, was sich in ihrer Zusammenfassung in RECORDs widerspiegelt.

Der Zugriff auf die einzelnen Bestandteile geschieht durch eine Selektion, die im Programm als *koordinate.x* notiert wird. Der Zugriff auf diesen RECORD-Bestandteil entspricht dem Umgang mit „normalen" Variablen.

Es ist nun möglich, einen RECORD in seiner Gesamtheit als Variable aufzufassen und mit ihm die Zuweisung auszuführen. Ein *o:=koordinate;* weist allen Bestandteilen von *o* den Inhalt des jeweils korrespondierenden Bestandteils in *koordinate* zu.

Man spricht auch von RECORD-Feldern, um einen Bestandteil zu bezeichnen. Dabei darf aber keine Konfusion mit den eigentlichen Feldern, den ARRAYs, auftreten. Mit Feldern meinen wir im Folgenden die Bestandteile eines RECORDs.

Für gewöhnliche RECORDs lassen sich sehr viele Beispiele finden - das beliebteste ist ein RECORD, der die Adresse eines Bekannten oder Geschäftspartners aufnehmen kann -, wir überspringen die Anwendungen und beschäftigen uns etwas näher mit der Speicherbelegung bei RECORDs.

Wie wird ein RECORD im Speicher abgelegt? Die erste Annahme, daß einfach die einzelnen Felder hintereinander abgelegt werden, stimmt nur im Prinzip.

Auf dem 68000 müssen Daten, die durch 16-Bit-Zugriffe manipuliert werden, auf geraden Adressen abgelegt sein. Dort, wo Byte-große Daten in einem RECORD stehen, müssen die nachfolgenden Daten eventuell durch Füll-Bytes auf gerade Speicheradressen gebracht werden. Nehmen wir einen RECORD *r1*:

```
VAR r1  :  RECORD
             a:BOOLEAN;
             b:INTEGER;
           END;
```

Wir gehen nun davon aus, daß BOOLEANs mit acht Bits implementiert sind. Sollte dies auf Ihrem System nicht der Fall sein, können Sie an deren Stelle einfach CHAR einsetzen. INTEGERs sollen 16 Bits, LONGCARDs 32 Bits groß sein - Annahmen, die von den meisten Compilern auf dem ST erfüllt werden.

Demnach beträgt die Summe der Speichergröße der Felder von *r1* drei Bytes. Befragen wir das System aber mit TSIZE, so sind für *r1* tatsächlich vier Bytes vorgesehen. Das zusätzliche Byte ist das Füll-Byte. Da alle Variablen immer an geraden Adressen ausgerichtet werden, läge *r1.b* ohne es an einer ungeraden Adresse. Also fügt der Compiler zwischen *r1.a* und *r1.b* ein - niemals genutztes - Byte ein.

Da alle Variablen an geraden Adressen beginnen müssen, kann es auch ein Füll-Byte nach dem letzten Feld geben. TSIZE liefert dementsprechend für das folgende *r2* eine Speichergröße von 6 Bytes.

```
VAR r2  :  RECORD
             a:BOOLEAN;
             b:INTEGER;
             c:BOOLEAN;
           END;
```

Da Variablen mit Byte-Größe aber auch an ungeraden Adressen angesprochen werden können, braucht zwischen Ihnen kein Füll-Byte eingefügt zu werden. *r2* ließe sich platzsparender als *r3* neuformulieren, für das nur noch vier Bytes gebraucht werden:

```
VAR r3  :  RECORD
             a,c:BOOLEAN;
             b  :INTEGER;
           END;
```

Sollten Sie Megamax-Modula benutzen, müßten Sie zur Überprüfung übrigens jetzt CHAR anstelle von BOOLEAN benutzen.

Die Moral von der Geschicht': Ist man in Speichernot, kann man durch geeignete Umordnung der Felder in RECORDs vielleicht noch einige Bytes herausschlagen. Beachten Sie aber, daß diese Methode vom verwendeten Prozessor abhängig ist - würde ein 68099 auch ungerade Wortzugriffe erlauben, gäbe es keinen Gewinn.

Es gibt übrigens Methoden im Compilerbau, die eine solche Umgruppierung automatisch vornehmen - auf dem ST werden sie aber leider nicht angewandt.

Keine Illusionen sollte man sich auch über die Möglichkeiten zur Packung von Unterbereichstypen machen. Ein RECORD

```
TYPE Zeit = RECORD
              Stunde : [0..23];
              Minute : [0..59];
            END;
```

könnte mit 11 Bits dargestellt werden. Tatsächlich wird die Größe des Typs genommen, aus dem der Unterbereich stammt. Damit benötigt *Zeit* vier Bytes, da es sich um zwei CARDINAL-Unterbereiche handelt.

Die im Modula-Vorgänger enthaltene Konzeption des gepackten - also platzsparend „zusammengeschobenen" - RECORDs gibt es in Modula nicht mehr.

ARRAY OF RECORD

Im Zusammenhang mit ARRAYs, die aus RECORDs bestehen, wird die WITH-Anweisung interessant. Sei ein *a* deklariert als ARRAY[0..100] OF Datum. *Datum* ist dabei wie *Zeit* deklariert und enthält die Felder *day*, *month* und *year*. Dann läßt sich mit *WITH a[i*3] DO ... END* festlegen, daß alle Namen innerhalb des WITH-Blocks sich auf die Felder des RECORDs beziehen. Eine Zuweisung *day:=1* darin verändert also eigentlich *a[i*3].day*. WITH ist eigentlich eher eine Compiler-Anweisung, da sie theoretisch nur die Bedeutung des Bezeichners im Programmtext verändert.

Tatsächlich bietet sie dem Compiler aber die Möglichkeit, eine kleine Optimierung vorzunehmen. Werden in einem WITH-Block mehrere Felder des RECORDs verändert, so läßt sich der Index bezogen auf das ARRAY einmalig vor dem WITH-Block errechnen. Damit spart man die wiederholten Indexberechnungen ein. Aus einem

```
WITH a[i*3] DO
  day:=1;
  month:=2;
  year:=1990;
END;
```

macht der Compiler also zur Optimierung die Codesequenz

```
tempind:=i*3;
a[tempind].day:=1;
a[tempind].month:=2;
a[tempind].year:=3;
```

Die Einsparung zweier langsamer Multiplikationen durch Einführung einer - compilerinternen - Zwischenvariablen *tempind* ist offensichtlich. Man sollte also aus Effizienzgründen bei ARRAYs, die aus RECORDs bestehen, die WITH-Anweisung einsetzen.

Hochoptimierende Compiler, die den gemeinsamen Ausdruck in allen Indizes der Version ohne WITH erkennen, sind auf „kleinen" Rechnern wie dem ST leider noch nicht verfügbar und benötigen sicherlich auch etwas mehr Rechenleistung.

Wir verlassen damit die gemeinen RECORDs und wenden uns ihren vielleicht interessanteren Spielarten zu: den varianten RECORDs.

Variante RECORDs

Normale RECORDs haben etwas Statisches, ihre Feldes sind unabänderlich festgelegt. Es gibt aber Anwendungen, in denen man Daten zu einem RECORD verbinden will, diese aber nicht immer aus denselben Feldern bestehen.

Stellen wir uns eine statistische Untersuchung über Verkehrsteilnehmer vor. Am Straßenrand steht ein Beamter, ausgestattet mit STACY, dem Laptop und einer Uhr. Er soll nun für jeden vorbeikommenden Verkehrsteilnehmer Daten aufnehmen, wobei natürlich jede Personenbezogenheit ausgeschlossen ist.

Er soll - in für seine Berufsgruppe gerüchteweise unüblicher Geschwindigkeit - eintippen, wann der Teilnehmer vorbeikam und dann je nach Fortbewegungsart vermerken, welchen Hubraum das Auto hat (er ist „Motorsport"-Experte!), ob es sich bei muskelbetriebenen Zweirädern um Tandems handelt oder nicht und wieviele Personen auf einem Motorrad sitzen (... bleiben - hallo Patrick!).

Das Programm, mit dem er arbeitet, stellt einen RECORD-Typ *Teilnehmer* bereit, in dem alle Daten vermerkt werden. Mit normalen RECORDs sähe das so aus:

```
TYPE Geraet = (Auto, Fahrrad,
               Motorrad);
  Teilnehmer = RECORD
               Uhrzeit  : Zeit;
               Art      : Geraet;
               Hubraum  : CARDINAL;
               Tandem   : BOOLEAN;
               Personen : CARDINAL;
              END;
```

Diese Lösung ist nicht effizient, denn immerhin bleiben jeweils zwei der fünf Felder ungenutzt - bei der Fahrradfahrerin beispielsweise bleibt die Personenanzahl unbelegt. Es handelt sich um eine hervorragende Gelegenheit für die Anwendung eines varianten RECORDs. Bei ihm werden in Abhängigkeit von einem Feld nur die jeweils benötigten anderen deklariert. Syntaktisch übernimmt ein varianter RECORD die Form einer CASE-Anweisung:

```
Teilnehmer = RECORD
               Uhrzeit  : Zeit;
               CASE Art : Geraet OF
      Auto     : Hubraum  : CARDINAL;
    | Fahrrad  : Tandem   : BOOLEAN;
    | Motorrad : Personen : CARDINAL;
               END
             END;
```

In *Teilnehmer* bestimmt das Feld *Art* das Aussehen der folgenden Felder. Enthält *Art Fahrrad*, so gilt das Feld *Tandem*, bei *Auto* ist *Hubraum* belegt. Es können also keine Felder unbelegt bleiben, da beispielsweise bei *Motorrad* keine anderen als *Personen* mehr existieren.

Gleichzeitig ist der Programmtext besser lesbar geworden, denn er sagt aus, daß für ein Auto nur der Hubraum von Interesse ist. Syntaktisch sollten Sie registrieren, daß *END* zweimal nötig ist - einmal zum Abschluß des CASEs und einmal zur Beendigung des RECORDs. Das Feld, von dem das CASE abhängt, nennt man übrigens auch *Tag*-Feld. Im Beispiel ist dies das Feld *Art*.

Für den Typ des Tag-Felds sind übrigens alle Aufzählungstypen möglich, also auch INTEGERs, CARDINALs oder BOOLEAN. Bei *Teilnehmer* handelt es sich um den selbstdefinierten Aufzählungstyp *Gerät*.

Wie sieht nun die Speicherbelegung aus? Eine Anfrage per TSIZE liefert das vielleicht überraschende Ergebnis 4. Tatsächlich sind ja Felder deklariert worden, die (zusammen mit einem Füll-Byte) insgesamt vierzehn Bytes belegen.

Die Erklärung ist einfach: In einem varianten RECORD werden die varianten Felder im selben Speicherplatz abgelegt. Da dieser für alle varianten Felder ausreichen muß, ist seine Größe immer die des größten Feldtyps. Bei

```
VAR r4 : RECORD CASE a:INTEGER OF
           0   : b:LONGCARD
         | 1   : c:INTEGER
         END END;
```

wäre der Speicherbedarf gleich sechs Bytes, nämlich zwei für das INTEGER *a* und vier von dem größten varianten Feld *b*.

Man könnte meinen, daß bei einer Belegung von *a* mit 1 das LONGCARD nicht ansprechbar wäre und bei 10 keine varianten Felder existierten. Dies stimmt nicht - der Compiler kann die Belegung nicht vorhersagen, und im Code verzichtet man auf Überprüfungen, da sie zu stark

verlangsamen würden. Man kann also durchaus eine Anweisungsfolge

```
r4.a:=1;
r4.c:=10000;
i:=r4.b;
```

notieren. Die bemerkenswerte Folge ist, daß man damit einen Typtransfer durchgeführt hat. Zunächst wird in *r4.c* das Bit-Muster für die INTEGER-Darstellung von 10000 geschrieben. Mit der Leseoperation erhält *i* den Inhalt von *r4.b*, der sich aus dem Bit-Muster von 10000 und einem undefinierten Wert zusammensetzt.

Der Compiler und das Laufzeitsystem helfen einem also nicht; die Verwendung eines varianten RECORDs wird daher immer von einem CASE zu steuern sein. Eine korrekte Routine zum Ausgeben des Inhalts von *r4* würde lauten:

```
...
  CASE r4.a OF
    0 :InOut.WriteLongcard(r4.b,10)
  | 1 :InOut.WriteInt(r4.b,6)
  END;
...
```

Listing 1 zeigt das Modul *Variables*, in dem ein Typ *Variable* deklariert wird, der verschiedene Zahlentypen aufnehmen kann. Er sollte nur durch die entsprechenden Zuweisungs- und Leseprozeduren manipuliert werden. *WriteVar* gibt eine solche Variable in Abhängigkeit von ihrem Typ mit den korrekten *InOut*-Prozeduren aus. An dieser Stelle müssen Sie eventuell Anpassungen an Ihr Modula-System vornehmen.

Variante RECORDs ohne Tag

Das Tag-Feld speichert die Information, welches der varianten Felder gültig ist. In *Variables* wird besonders deutlich, daß diese Information nötig ist. Ohne das Feld *Typ* wäre die typgerechte Ausgabe in *WriteVar* nicht möglich.

Der oben schon angesprochene implizite Typtransfer, der über einen varianten RECORD vorgenommen werden kann, bietet die Möglichkeit, einige Tricks zu implementieren. Will man einen varianten RECORD nur dafür verwenden, benötigt man das Tag-Feld eigentlich nicht. Es könnte auch sein, daß sich die Gültigkeit der varianten Felder aus anderen Daten außerhalb des RECORDs ergibt.

In diesem Fall kann man das Tag-Feld weglassen:

```
VAR r5 : RECORD CASE : INTEGER OF
           0    : b:LONGCARD
         | 1    : c:INTEGER
         END END;
```

Syntaktisch fällt einfach der Bezeichner des Tag-Felds weg. Da das verstümmelte CASE allerdings immer noch einen Typ für die einzelnen Varianten benötigt, muß der Typbezeichner übrigbleiben. Der etwas einsam stehende Doppelpunkt ermöglicht dem Compiler eine einfache Erkennung eines Tag-losen RECORDs.

Nun kann man wie schon bei *r4* auf *r5.b* als LONGCARD und *r5.c* als INTEGER zugreifen und beschreibt dabei natürlich wie bei allen varianten RECORDs denselben Speicherbereich.

Der Vorteil liegt darin, daß kein Speicher mehr für das Tag-Feld belegt wird. Und in der Tat belegt *r5* nur noch vier anstelle von sechs Bytes bei *r4*. Man kann nun also die varianten Felder für den gewünschten Typtransfer benutzen, ohne den überflüssigen Ballast eines Tag-Felds mitzuschleppen.

Vielleicht ließe sich argumentieren, daß Typtransfers ja schon durch SYSTEM.VAL möglich sind und diese - wie in der ersten Folge beschrieben - systemabhängige Annahmen sehr schön dokumentieren. VAL hat allerdings den Nachteil, daß es sich in der Regel auf Transfers zwischen einfachen Typen beschränkt.

Mit varianten RECORDs kann man zwischen allen denkbaren Typkonstruktionen konvertieren - oder besser gesagt wandeln, da ja nur das Bit-Muster übernommen und keine Anpassung vollzogen wird.

Ein Beispiel dazu ist die folgende Prozedur *DumpReal*, die eine Hex-Darstellung der einzelnen Bytes einer REAL-Variablen ausgeben soll. Wir nehmen dabei an, daß REALs mit vier Bytes dargestellt werden und BYTE und VAL aus SYSTEM importiert wurden.

```
PROCEDURE DumpReal(r:REAL);
VAR transfer  : RECORD CASE
                  :BOOLEAN OF
       FALSE  : r:REAL;
     | TRUE   : b:ARRAY[0..3] OF
BYTE
     END END;
    i         : INTEGER;
BEGIN
  transfer.r:=r;
  FOR i:=0 TO 3 DO
    InOut.WriteHex(VAL(CARDINAL,
                   transfer.b[i]),2)
  END
END DumpReal;
```

Die Prozedur greift über das variante BYTE-Feld auf die Speicherdarstellung des REAL-Werts zu. Die Möglichkeiten dieses Tricks sind enorm: Man kann prinzipiell immer zwischen allen Daten in jeder denkbaren Struktur umwandeln. Und dies ist enorm einfach möglich und beschränkt sich natürlich nicht nur auf den Tag-Typ BOOLEAN. Nehmen Sie CARDINAL, und Sie haben 65336 verschiedene Darstellungen im Zugriff. In *DumpReal* scheint es vielleicht, daß solche Anwendungen nicht unbedingt effizient sind, denn immerhin ist die Zuweisung des Parameters an den lokalen varianten RECORD nötig. Aufgewogen wird dies allerdings dadurch, daß bei keinem Lesezugriff zusätzlicher Aufwand entsteht.

Natürlich sind solche Anwendungen zutiefst systemabhängig und leider nur durch Kommentare als solche markierbar. Die Darstellung eines Fließkommawertes mag vielleicht zwischen den Rechnersystemen durch die IEEE-Standards relativ ähnlich sein. Bei einem varianten RECORD, in dem eine Variante wiederum ein RECORD ist, muß man sich aber schon sehr gründlich überlegen, ob die oben genannten Füll-Bytes entstehen oder nicht.

Abschließend in Listing 2 noch zwei Prozeduren, die für verschiedene Anwendungen interessant sein können - sie ermöglichen den Zugriff auf die höheren und niederen Bytes von 16-Bit-Variablen. Das Ergebnis der Prozeduren muß wie bei *BoolWordOps* mit VAL auf den gewünschten Typ gebracht werden.

Für Sie bleibt die kleine Aufgabe - entsprechend den Möglichkeiten, Langwörter in Abhängigkeit von Ihrem SYSTEM-Modul zu verwenden - entsprechende Routinen für den Zugriff auf die oberen und unteren Wörter von Langwort-Parametern (LONGCARD, LONGINT etc.) zu implementieren.

RT

```
 1:  DEFINITION MODULE Variables;
 2:  (* Robert Tolksdorf fuer ST-Computer, 8/90 *)
 3:
 4:  TYPE VariableTypes = (Boolean, Integer, Cardinal,
                           Longint, Longcard,
 5:                        Real, Longreal);
 6:       Variable      = RECORD CASE
                          Typ:VariableTypes OF
 7:                         Boolean  : b  : BOOLEAN;
 8:                       | Integer  : i  : INTEGER;
 9:                       | Cardinal : c  : CARDINAL;
10:                       | Longint  : li : LONGINT;
11:                       | Longcard : lc : LONGCARD;
12:                       | Real     : r  : REAL;
13:                       | Longreal : lr : LONGREAL;
14:                       END END;
15:
16:  PROCEDURE SetBool(VAR v:Variable; val:BOOLEAN);
17:  PROCEDURE BoolValue(v:Variable):BOOLEAN;
18:  PROCEDURE SetInt(VAR v:Variable; val:INTEGER);
19:  PROCEDURE IntValue(v:Variable):INTEGER;
20:  PROCEDURE SetCard(VAR v:Variable; val:CARDINAL);
21:  PROCEDURE CardValue(v:Variable):CARDINAL;
22:  PROCEDURE SetLongint(VAR v:Variable; val:LONGINT);
23:  PROCEDURE LongintValue(v:Variable):LONGINT;
24:  PROCEDURE SetLongcard(VAR v:Variable;
                           val:LONGCARD);
25:  PROCEDURE LongcardValue(v:Variable):LONGCARD;
26:  PROCEDURE SetReal(VAR v:Variable; val:REAL);
27:  PROCEDURE RealValue(v:Variable):REAL;
28:  PROCEDURE SetLongreal(VAR v:Variable;
                           val:LONGREAL);
29:  PROCEDURE LongrealValue(v:Variable):LONGREAL;
30:  PROCEDURE WriteValue(v:Variable);
31:  END Variables.
```

```
 1:  IMPLEMENTATION MODULE  Variables;
 2:  (* Robert Tolksdorf fuer ST-Computer, 8/90 *)
 3:  (* Hier sind eventuell Anpassungen noetig ! *)

 4:  FROM InOut      IMPORT WriteString, WriteInt,
                            WriteCard;
 5:  FROM LongInOut  IMPORT WriteLongInt,
                            WriteLongCard;
 6:  FROM RealInOut  IMPORT WriteReal, WriteLongReal;
 7:
 8:  (* Vom Definitions-Modul:
 9:  TYPE VariableTypes = (Boolean, Integer, Cardinal,
                           Longint, Longcard,
10:                        Real, Longreal);
11:       Variable      = RECORD CASE
                          Typ:VariableTypes OF
12:                         Boolean  : b  : BOOLEAN;
13:                       | Integer  : i  : INTEGER;
14:                       | Cardinal : c  : CARDINAL;
15:                       | Longint  : li : LONGINT;
16:                       | Longcard : lc : LONGCARD;
17:                       | Real     : r  : REAL;
18:                       | Longreal : lr : LONGREAL;
19:                       END END;
20:  *)
21:
22:  PROCEDURE SetBool(VAR v:Variable; val:BOOLEAN);
23:  BEGIN
24:    v.Typ:=Boolean;
25:    v.b:=val
26:  END SetBool;
27:
28:  PROCEDURE BoolValue(v:Variable):BOOLEAN;
29:  BEGIN
30:    IF v.Typ=Boolean THEN
31:      RETURN v.b
32:    END
33:  END BoolValue;
34:
35:  PROCEDURE SetInt(VAR v:Variable; val:INTEGER);
36:  BEGIN
37:    v.Typ:=Integer;
38:    v.i:=val
39:  END SetInt;
40:
41:  PROCEDURE IntValue(v:Variable):INTEGER;
42:  BEGIN
43:    IF v.Typ=Integer THEN
44:      RETURN v.i
45:    END
46:  END IntValue;
47:
48:  PROCEDURE SetCard(VAR v:Variable; val:CARDINAL);
49:  BEGIN
50:    v.Typ:=Cardinal;
51:    v.c:=val
52:  END SetCard;
53:
54:  PROCEDURE CardValue(v:Variable):CARDINAL;
55:  BEGIN
56:    IF v.Typ=Cardinal THEN
57:      RETURN v.c
58:    END
59:  END CardValue;
60:
61:  PROCEDURE SetLongint(VAR v:Variable;val:LONGINT);
62:  BEGIN
63:    v.Typ:=Longint;
64:    v.li:=val
65:  END SetLongint;
66:
67:  PROCEDURE LongintValue(v:Variable):LONGINT;
68:  BEGIN
69:    IF v.Typ=Longint THEN
70:      RETURN v.li
71:    END
72:  END LongintValue;
73:
74:  PROCEDURE SetLongcard(VAR v:Variable;
                           val:LONGCARD);
75:  BEGIN
76:    v.Typ:=Longcard;
77:    v.lc:=val
78:  END SetLongcard;
79:
80:  PROCEDURE LongcardValue(v:Variable):LONGCARD;
81:  BEGIN
82:    IF v.Typ=Longcard THEN
83:      RETURN v.lc
84:    END
85:  END LongcardValue;
86:
87:  PROCEDURE SetReal(VAR v:Variable; val:REAL);
88:  BEGIN
89:    v.Typ:=Real;
90:    v.r:=val;
91:  END SetReal;
92:
93:  PROCEDURE RealValue(v:Variable):REAL;
94:  BEGIN
95:    IF v.Typ=Real THEN
96:      RETURN v.r
97:    END;
98:  END RealValue;
99:
100: PROCEDURE SetLongreal(VAR v:Variable;
                           val:LONGREAL);
101: BEGIN
102:   v.Typ:=Longreal;
103:   v.lr:=val
104: END SetLongreal;
105:
106: PROCEDURE LongrealValue(v:Variable):LONGREAL;
107: BEGIN
108:   IF v.Typ=Longreal THEN
109:     RETURN v.lr
110:   END
111: END LongrealValue;
112:
113: PROCEDURE WriteValue(v:Variable);
114: BEGIN
115:   CASE v.Typ OF
116:     Boolean  : IF v.b THEN WriteString('TRUE')
117:                       ELSE WriteString('FALSE')
118:                END
119:   | Integer  : WriteInt(v.i,6)
120:   | Cardinal : WriteCard(v.c,5);
121:   | Longint  : WriteLongInt(v.li,11);
122:   | Longcard : WriteLongCard(v.lc,10);
123: (* Die Stellenanzahl der REAL/LONGREAL-Ausgaben
      anpassen! *)
124:   | Real     : WriteReal(v.r,15);
125:   | Longreal : WriteLongReal(v.lr,15);
126:   ELSE
127:     HALT
128:   END
129: END WriteValue;
130: END Variables.
```

Compiler-Bau



Viele Programmierer sehen in einem Compiler immer noch ein fast mystisches Werkzeug, das die von ihnen geschriebenen Programme auf wundersame Weise von einer Hochsprache, sei es Modula-2, C, Pascal oder auch BASIC, in Maschinencode übersetzt. Doch auch ein Compiler ist nur ein ganz normales Programm, dessen Funktionsweise jeder verstehen kann. Gerade im Compiler-Bau wurden in der Informatik in den letzten Jahren große Fortschritte erzielt, so daß aus den einzelnen Tricks und Kniffen der verschiedenen Compiler-Bauer inzwischen eine fast vollständige Sammlung von Programmiermethoden geworden ist.

Teil 1

In fünf Artikeln werden wir uns durch das Innenleben eines Compilers kämpfen. Dabei werden wir die einzelnen Aufgaben eines Compilers Schritt für Schritt kennenlernen und gleichzeitig sehen, wie diese Aufgaben gelöst werden können. Die dabei erworbenen Kenntnisse sind für jeden Programmierer interessant, da er mit ihrer Hilfe bessere Vergleiche zwischen verschiedenen Compilern anstellen kann und einige der Techniken auch in anderen Programmen Verwendung finden. Außerdem fällt einem der Umgang mit einem Programm, dessen Funktionsweise man versteht, in der Regel um einiges leichter. Vielleicht gewinnt der eine oder andere sogar so viel Interesse an dem Thema Compiler-Bau, daß er eines Tages seinen eigenen Compiler schreibt.

Überblick

In dem Rest dieser Folge werden wir uns einen allgemeinen Überblick über die einzelnen Komponenten eines Compilers, ihre Aufgaben und ihr Zusammenspiel verschaffen. Das nächste Mal werden wir zuerst ein wenig vom eigentlichen Thema abweichen und uns mit der Laufzeitumgebung eines compilierten Programms befassen. Danach beschäftigen wir uns mit der ersten und einfachsten Phase eines Compilers, der lexikalischen Analyse, auch Scanner genannt. In der dritten Folge wollen wir uns den Parser, also die syntaktische Analyse, etwas genauer ansehen; und da dies ein recht umfangreiches Thema ist, werden wir auch den gesamten Artikel dazu benötigen. In der vierten Folge behandeln wir dann die semantische Analyse. Sie bildet den Abschluß der Programmanalyse, also des Teils eines Compilers, der ein Programm in seine Einzelteile zerlegt und deren Bedeutung analysiert. Danach beginnen wir mit der Besprechung des Syntheseteils, der dann schlußendlich den Maschinencode erzeugt. Dabei befassen wir uns zunächst mit der Erzeugung eines internen Zwischencodes. In der fünften und letzten Folge wollen wir uns zuerst die verschiedenen Arten der endgültigen Codererzeugung ansehen, und danach reißen wir kurz das Thema Codeoptimierung an. Zu guter Letzt werden wir uns dann eine kleine Beispielsprache anschauen, wobei jeder aufgerufen ist, mit Hilfe des inzwischen erworbenen Wissens einen kleinen Compiler für diese Sprache zu schreiben.

Was gibt's zu tun?

Die allgemein bekannte Aufgabe eines Compilers ist es, ein in einer Hochsprache verfaßtes Programm (evtl. auch einen Text, siehe TeX) in eine Form umzuwandeln, die der Computer ausführen (bzw. drukken) kann. Dazu muß der Compiler das Programm zuerst analysieren, d.h. das Programm wird in seine einzelnen Bestandteile, wie Konstanten-, Typ-, Variablen- und Prozedurdeklarationen zerlegt. Genau genommen werden auch diese Bestandteile noch in einzelne Anweisungen wie Schleifen, Zuweisungen, Prozeduraufrufe usw. zerteilt. Dabei wird auch alle für den Computer überflüssige Information wie Kommentare und Formatierung entfernt. Sogar die Namen der einzelnen Bezeichner sind für ihn uninteressant, es genügt, sie durchzunumerieren und fortan nur noch mit den Nummern zu arbeiten (was übrigens auch schneller geht, da der Vergleich zweier Zeichenketten sehr viel aufwendiger ist als der von zwei Zahlen). Bei diesem Zerteilen erkennt der Compiler auch alle eventuell vorhandenen grammatikalischen Fehler, wie zum Beispiel falsche Klammerung oder die allseits bekannten vergessenen Strichpunkte.

Nachdem das Programm derart zerstückelt ist, versucht der Compiler, einen Teil der Bedeutung des Programms zu verstehen. Aus der Anweisung $c := a + b$ ersieht man zwar sofort, daß es sich um eine Addition handelt, es ist aber unklar, ob zwei Integer- oder zwei Fließkommazahlen addiert werden. Um das zu ermit-

teln, muß der Compiler auch andere Teile des Programms heranziehen. In unserem Beispiel die Deklaration der einzelnen Variablen, die etwa *VAR a,b,c: INTEGER;* lauten könnte und klarstellt, daß es sich um eine Addition zweier Integer-Zahlen handelt. Solche Schlußfolgerungen sind Teil der Analyse der statischen Semantik. Semantik ist das griechische Wort für Bedeutung, und mit statischer Semantik bezeichnet man den Teil der Bedeutung eines Programms, der vom Computer errechnet werden kann, ohne daß das Programm gestartet werden muß. Zum Beispiel gehört die Tatsache, daß ein Programm für eine gewisse Eingabe in einer Endlosschleife hängenbleibt, nicht zur statischen, sondern zur dynamischen Semantik.

Für den Compiler ist das Ermitteln der statischen Semantik wichtig, da er mit ihrer Hilfe den passenden Code erzeugen kann. Man denke etwa an das obige Beispiel, in dem zwischen einer Ganzzahl- und einer Fließkommaddition unterschieden werden muß. Außerdem erkennt er dabei auch eine Reihe von Programmierfehlern, wie zum Beispiel Typkonflikte oder die falsche Anzahl an Parametern in einem Prozedur- oder Funktionsaufruf.

Nachdem der Compiler das Programm derart zerlegt und analysiert hat, ist er bereit, seinerseits ein Programm in der Zielsprache zu schreiben, dessen Funktion mit dem im Quelltext beschriebenen Programm identisch ist. Dazu wird das Programm manchmal zuerst in einen Zwischencode übersetzt, der der Zielsprache sehr ähnlich ist, aber noch möglichst viele Details versteckt, wie zum Beispiel das genaue Vorgehen beim Berechnen des Speicherplatzes eines Array-Elements. Dieser Zwischencode kann nun optimiert werden, d.h. es können zum Beispiel konstante Ausdrücke berechnet und durch ihren Wert ersetzt oder mehrfach durchgeführte gleichartige Berechnungen eliminiert werden. Der optimierte Zwischencode wird schlußendlich in den eigentlichen Zielcode übersetzt, und es werden eventuell noch Tabellen für einen Linker oder einen Debugger hinzugefügt.

Recycling

Da die gerade beschriebenen Komponenten in verschiedenen Compilern oft sehr ähnlich sind, stellt sich natürlich auch die Frage der Wiederverwendbarkeit. Soll zum Beispiel ein Modula-2-Compiler für den Atari ST und den Amiga geschrieben werden, können beide Compiler fast vollkommen gleich aussehen, da beide Rechner mit einer 680x0 arbeiten und daher sogar die Codeerzeugung gleich ist. Nur die



Betriebssystemaufrufe, die zum Beispiel zum Einlesen des Quellcodes gebraucht werden, müssen verschieden sein. Normalerweise sollten alle betriebssystemabhängigen Routinen sowieso in einem speziellen Modul zusammengefaßt werden, so daß in diesem Fall nur ein einziges Modul geändert werden muß. Schwieriger ist es, einen Compiler auf einem ST und einem MS-DOS-Rechner zu verwenden, da beide Rechner unterschiedliche Mikroprozessoren besitzen. Da Scanner und Parser von einer derartigen Umstellung aber völlig unberührt bleiben, können diese beiden Komponenten wiederverwendet werden. Theoretisch ist lediglich die Codeerzeugung verschieden, so daß es genügen würde, sie umzuschreiben. Leider ist dies in Wirklichkeit oft nur Theorie, da die Codeerzeugung meist Rückwirkungen auf die semantische Analyse hat. Trotzdem gibt es einige Beispiele, in denen es gelungen ist, einen gut modularisierten Compiler recht einfach auf teilweise sehr verschiedene Rechner zu portieren.

Compilercompiler

Ein weiterer Ansatz, um Arbeit beim Schreiben von Compilern zu sparen, sind die sogenannten Compilergeneratoren, auch Compilercompiler genannt. Der berühmteste seiner Art ist wohl YACC (Yet Another Compiler Compiler), der zusammen mit seinem Partner LEX (LEXical analyser) die komplette syntaktische Analyse eines Compilers erzeugen kann. Beide Programme laufen unter UNIX und können beim Schreiben eines Compilers eine große Hilfe sein. Leider erzeugen die meisten Compilergeneratoren keine kompletten Compiler, sondern nur einzelne Komponenten, so daß für den Programmierer immer noch genug zu tun bleibt. Außerdem nehmen einem solche Werkzeuge nur die eigentliche Codierungsarbeit ab, die Konzeption und das Erstellen der Eingabedateien für die Generatoren ist nach wie vor dem Menschen überlassen. Am Ende der Besprechung einer jeden Compilerkomponente werden wir sehen, welche Arbeiten daran von einem Compilergenerator übernommen werden können.

Arbeitskollegen des Compilers

Bevor wir uns jetzt etwas genauer mit dem Aufbau eines Compilers befassen, werfen wir noch einen Blick auf die Programme eines Entwicklungssystems, die direkt mit dem Compiler zusammenarbeiten müssen.

Der erste Schritt beim Codieren eines Programms ist das Eingeben des Quelltextes mit Hilfe eines Editors. In der Regel ist es dem Editor recht egal, welcher Compiler das Programm später übersetzt oder in welcher Programmiersprache das Programm geschrieben wurde. Eine Ausnahme bilden hier die sogenannten Struktur-Editoren. Ein Struktur-Editor prüft schon bei der Eingabe die grammatikalische Korrektheit des Programms und läßt nur korrekte Eingaben zu. In einem integrierten System kann er dem Compiler sogar die syntaktische Analyse des Programms ersparen, indem er ihm das Programm gleich in analysierter Form übergibt.

Oft sind ein Präprozessor und ein Assembler direkt in den Compiler integriert. Ein Präprozessor kann zum Beispiel für Makrodeklarationen, bedingte Compilierung und das Einfügen anderer Quelltexte in das zu compilierende Programm verwendet werden. Ein Assembler übersetzt den vom Compiler erzeugten, symbolischen Assembler-Code in den eigentlichen Maschinencode. Dabei wird der Compiler in der Regel natürlich keine Textdatei mit dem symbolischen Assembler-Code an den integrierten Assembler übergeben, sondern die Daten werden als interne Datenstruktur irgendwo im Speicher liegen. Oder der Assembler wird sogar „on the fly" aufgerufen, d.h. sobald der Compiler einen symbolischen Assembler-Befehl erzeugt hat, ruft er eine Assembler-Routine mit diesem Befehl auf. Diese wiederum verarbeitet die eine Anweisung und kehrt dann sofort zurück, um die nächste Anweisung zu erhalten. Früher, als die Rechner noch sehr viel weniger Speicherplatz hatten, waren Präprozessor, Compiler und Assembler oft getrennte Programme, und zur Kommunikation diente ihnen nur der externe Massenspeicher. Dadurch mußte immer nur eine Komponente im Speicher sein, was natürlich Platz spart. Dieses Verfahren hat allerdings den Nachteil, daß es recht langsam ist, deswegen packt man heute lieber alles in ein Programm.

Da man ein Programm in der Regel nicht nur für einen bestimmten Speicherbereich des Computers schreibt und meist noch irgendwelche Bibliotheken eingebunden werden müssen, ist es mit dem Assem-

blieren eines Programms immer nocht nicht getan. Um das Programm ausführen zu können, muß es noch reloziert werden, d.h. alle Zugriffe auf absolute Adressen müssen auf den Speicherbereich angepaßt werden, in dem das Programm schlußendlich läuft. Außerdem müssen alle Aufrufe von Bibliotheksfunktionen aufgelöst werden, d.h. es muß die Startadresse der entsprechenden Funktion eingesetzt werden. Dies alles wird von einem Linker und/oder Loader erledigt. Dieser verrichtet seine Arbeit in der Regel anhand einer Reloziertabelle, die vom Compiler in den Programmcode eingefügt wird.

Gerecht geteilt

Wie wir oben schon gesehen haben, kann man einen Compiler ganz grob in zwei Teile zerlegen. In den Analyseteil, der das Quellprogramm zerlegt und analysiert, und in den Syntheseteil, der aus der gewonnenen Information ein äquivalentes, also funktionsgleiches Programm in der Zielsprache erzeugt. Diese beiden Teile werden als „front end" und „back end" bezeichnet und sind, wenn sie durch eine saubere Schnittstelle getrennt werden, recht unabhängig voneinander.

Natürlich gibt es auch Compilerkomponenten, die weder dem einen noch dem anderen Teil zugeordnet werden können. Zum Beispiel die Verwaltung der Symboltabelle und die Fehlerbehandlung, da beides sowohl im Front-End als auch im Back-End gebraucht wird.

Das Front-End hängt im wesentlichen von den Eigenschaften der Quellsprache ab, und das Aussehen des Back-Ends wird hauptsächlich vom Zielcode bestimmt.

Autopsie

Nach diesen einleitenden Worten sehen wir uns den Aufbau eines Compilers mal etwas genauer anhand von Abb. 1 an. Bei den rechteckigen Kästen handelt es sich um die wichtigsten Programmkomponenten und bei den Ellipsen um die verschiedenen Formen, die das Programm während seiner Transformation vom Quelltext zum Maschinencode durchläuft. Die ersten drei Rechtecke, die mit Scanner, Parser und semantische Analyse bezeichnet sind, bilden zusammen den Analyseteil oder auch das Front-End des Compilers. Die Zwischencodeerzeugung, Codeoptimierung und Codeerzeugung werden zusammen Syntheseteil oder auch Back-End genannt. Auf der rechten Seite befinden sich die drei wichtigsten globalen Tabellen, dies sind die Bezeichner-, Definitions- und Konstantentabelle, die wir später noch genauer unter die Lupe nehmen wollen. Außerdem befindet sich dort die Fehlerbehandlung, die wie die drei Tabellen von allen Phasen des Compilers gemeinsam benutzt wird.

Selbstverständlich ist nicht jeder existierende Compiler genau nach diesem Schema aufgebaut. Vielen Compilern fehlt zum Beispiel die Codeoptimierungsphase vollständig, bei einfachen Compilern ist die Codeerzeugung unter Umständen derart in die semantische Analyse verwoben, daß eine Trennung der beiden fast unmöglich ist. Doch hat sich im Laufe der Zeit herausgestellt, daß recht viele Compiler nach diesem Schema aufgebaut sind und es bei der Entwicklung eines neuen Compilers meist von Vorteil ist, sich an das vorgestellte Konzept zu halten.

Metamorphose

Das zu übersetzende Programm wird von jeder Phase des Compilers in eine andere Form umgewandelt, bis es schlußendlich zum Zielcode wird. Wir wollen uns die einzelnen Schritte dieser Metamorphose nun etwas genauer ansehen.

Zuerst ist da der Scanner, der die sogenannte lexikalische Analyse durchführt. Seine Aufgabe ist es, alle unwichtigen Details aus dem Quelltext zu entfernen; dazu gehören alle Kommentare, in den meisten Programmiersprachen auch die Formatierung, und die genauen Namen der Bezeichner sowie die Darstellung der verschiedenen Konstanten. Wie schon erwähnt, werden statt der Bezeichnernamen im folgenden nur noch Zahlen verwendet, die den vorkommenden Bezeichnern am Anfang eindeutig zugeordnet werden. Wie das genau vor sich geht, legt die Bezeichnertabelle fest und wird uns später noch beschäftigen. Etwas Ähnliches geschieht mit den Konstanten, von denen uns zunächst nur einmal interessiert, ob es eine Zahl, eine Zeichenkette oder etwas anderes ist. Nicht aber der genaue Wert, er wird stattdessen in der Konstantentabelle abgelegt. Nachdem alles Unwichtige eliminiert ist, faßt der Scanner die verbliebene Information zu kleinen Päckchen, den Tokens, zusammen. D.h. statt dem Parser mitzuteilen, daß er soeben das Wort *PROCEDURE* gelesen hat, und dabei Buchstabe für Buchstabe zu übermitteln, schickt er einfach nur einen Code, der besagt, daß er das Schlüsselwort PROCEDURE erkannt hat. Genauso meldet er bei einem Bezeichner nur das Auftreten eines Bezeichners und dessen Nummer, wobei zwei gleichgeschriebene Bezeichner natürlich die gleiche Nummer erhalten. Alle Tokens eines Programms zusammengenommen bilden die Token-Folge des Programms. In ihr ist alle für den weiteren Übersetzungsvorgang nötige Information in sehr viel kompakterer Form als im Quelltext enthalten.

Die Token-Folge wird nun vom Parser, der für die grammatikalische Analyse zuständig ist, in einen sogenannten Strukturbaum umgewandelt. Wie man dem Namen entnehmen kann, ist dies keine lineare Struktur mehr, wie es die Token-Folge oder das Originalprogramm waren, sondern eine baumartige Struktur. Strukturbaum heißt das Ding, weil es die grammatikalische Struktur des Programms aufzeigt. Jetzt kann man sich natürlich fragen, was Modula-2 mit Grammatik zu tun hat: nach Deutsch, Englisch, vielleicht auch noch Latein oder Französisch hielt man dieses traurige Kapitel des eigenen Lebens doch für abgeschlossen. Doch auch eine Computersprache besitzt eine Grammatik. Genau wie der Satz einer natürlichen Sprache aus Subjekt, Objekt, Prädikat usw. aufgebaut ist, besteht ein Programm aus Konstanten-, Typ-, Variablen- und Prozedurdeklarationen sowie aus Zuweisungen, Prozeduraufrufen und vielem mehr. Genau diese Komponenten eines Programms erkennt der Parser und gruppiert sie in einem Baum. Dabei repräsentiert die Wurzel des Baumes das komplette Programm. Ihre Nachfolger sind die einzelnen Dekla-

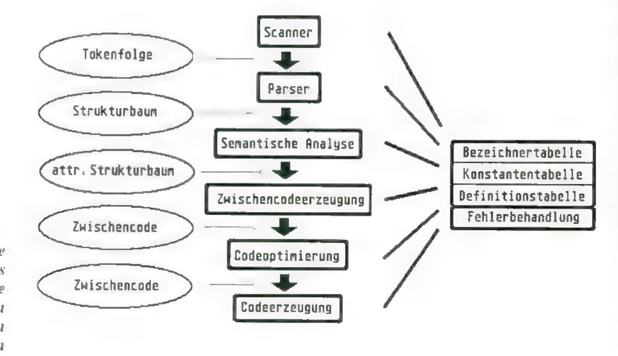
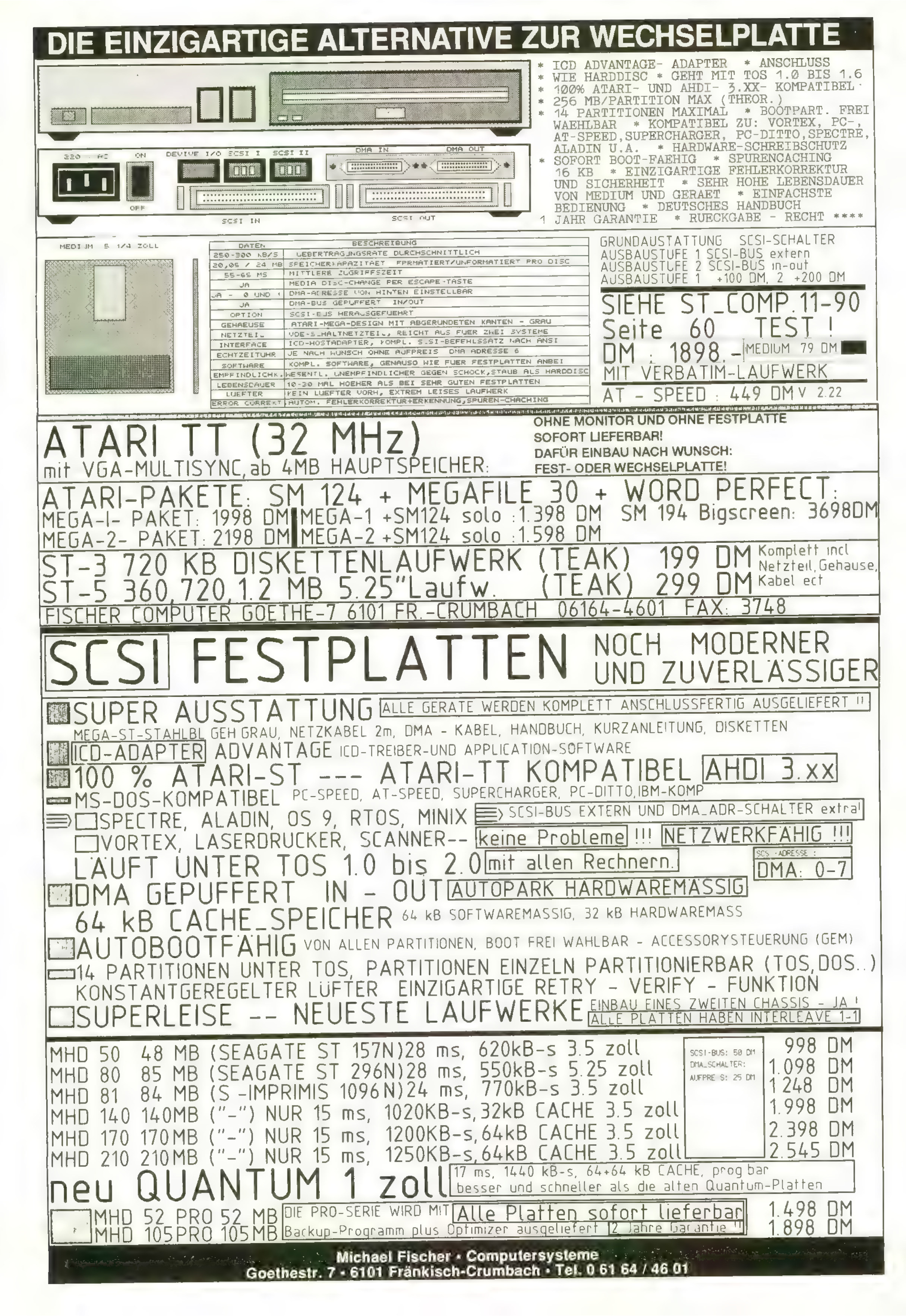


Abb. 1: Die Phasen eines Compilers und die von ihnen erzeugten Zwischenformen



DIE EINZIGARTIGE ALTERNATIVE ZUR WECHSELPLATTE
* ICD ADVANTAGE- ADAPTER * ANSCHLUSS WIE HARDDISC * GEHT MIT TOS 1.0 BIS 1.6
* 100% ATARI- UND AHDI- 3.XX- KOMPATIBEL
* 256 MB/PARTITION MAX (THEOR.)
* 14 PARTITIONEN MAXIMAL * BOOTPART. FREI WAEHLBAR * KOMPATIBEL ZU: VORTEX, PC-, AT-SPEED, SUPERCHARGER, PC-DITTO, SPECTRE, ALADIN U.A. * HARDWARE-SCHREIBSCHUTZ
* SOFORT BOOT-FAEHIG * SPURENCACHING 16 KB * EINZIGARTIGE FEHLERKORREKTUR UND SICHERHEIT * SEHR HOHE LEBENSDAUER VON MEDIUM UND GERAET * EINFACHSTE BEDIENUNG * DEUTSCHES HANDBUCH
1 JAHR GARANTIE * RUECKGABE - RECHT ****
220 AC
ON
OFF
DEVICE I/O
SCSI I
SCSI II
DMA IN
DMA OUT
SCSI IN
SCSI OUT
MEDIUM 5 1/4 ZOLL
DATEN | BESCHREIBUNG
250-300 KB/S | UEBERTRAGUNGSRATE DURCHSCHNITTLICH
20,05 / 24 MB | SPEICHERKAPAZITAET FORMATIERT/UNFORMATIERT PRO DISC
55-65 MS | MITTLERE ZUGRIFFSZEIT
JA | MEDIA DISC-CHANGE PER ESCAPE-TASTE
JA - 0 UND 1 | DMA-ADRESSE VON HINTEN EINSTELLBAR
JA | DMA-BUS GEPUFFERT IN/OUT
OPTION | SCSI-BUS HERAUSGEFUEHRT
GEHAEUSE | ATARI-MEGA-DESIGN MIT ABGERUNDETEN KANTEN - GRAU
NETZTEIL | VDE-SCHALTNETZTEIL, REICHT AUS FUER ZWEI SYSTEME
INTERFACE | ICD-HOSTADAPTER, KOMPL. SCSI-BEFEHLSSATZ NACH ANSI
ECHTZEITUHR | JE NACH WUNSCH OHNE AUFPREIS DMA ADRESSE 6
SOFTWARE | KOMPL. SOFTWARE, GENAUSO WIE FUER FESTPLATTEN ANBEI
EMPFINDLICHK. | WESENTL. UNEMPFINDLICHER GEGEN SCHOCK, STAUB ALS HARDDISC
LEBENSDAUER | 10-30 MAL HOEHER ALS BEI SEHR GUTEN FESTPLATTEN
LUEFTER | KEIN LUEFTER VORH, EXTREM LEISES LAUFWERK
ERROR CORREKT | AUTOM. FEHLERKORREKTUR+ERKENNUNG, SPUREN-CHACHING
GRUNDAUSTATTUNG SCSI-SCHALTER
AUSBAUSTUFE 1 SCSI-BUS extern
AUSBAUSTUFE 2 SCSI-BUS in-out
AUSBAUSTUFE 1 +100 DM, 2 +200 DM
SIEHE ST_COMP. 11-90
Seite 60 TEST !
DM : 1898.-
MEDIUM 79 DM
MIT VERBATIM-LAUFWERK
AT - SPEED : 449 DM V 2.22
ATARI TT (32 MHz)
mit VGA-MULTISYNC, ab 4MB HAUPTSPEICHER:
OHNE MONITOR UND OHNE FESTPLATTE
SOFORT LIEFERBAR!
DAFÜR EINBAU NACH WUNSCH:
FEST- ODER WECHSELPLATTE!
ATARI-PAKETE: SM 124 + MEGAFILE 30 + WORD PERFECT:
MEGA-1- PAKET: 1998 DM
MEGA-2- PAKET: 2198 DM
MEGA-1 +SM124 solo : 1.398 DM
MEGA-2 +SM124 solo : 1.598 DM
SM 194 Bigscreen: 3698DM
ST-3 720 KB DISKETTENLAUFWERK (TEAK) 199 DM
ST-5 360, 720, 1.2 MB 5.25"Laufw. (TEAK) 299 DM
Komplett incl Netzteil, Gehause, Kabel ect
FISCHER COMPUTER GOETHE-7 6101 FR.-CRUMBACH 06164-4601 FAX: 3748
SCSI FESTPLATTEN
NOCH MODERNER UND ZUVERLÄSSIGER
SUPER AUSSTATTUNG
ALLE GERATE WERDEN KOMPLETT ANSCHLUSSFERTIG AUSGELIEFERT !!
MEGA-ST-STAHLBL GEH GRAU, NETZKABEL 2m, DMA - KABEL, HANDBUCH, KURZANLEITUNG, DISKETTEN
ICD-ADAPTER ADVANTAGE ICD-TREIBER-UND APPLICATION-SOFTWARE
100 % ATARI-ST --- ATARI-TT KOMPATIBEL
AHDI 3.xx
MS-DOS-KOMPATIBEL PC-SPEED, AT-SPEED, SUPERCHARGER, PC-DITTO, IBM-KOMP
SPECTRE, ALADIN, OS 9, RTOS, MINIX
SCSI-BUS EXTERN UND DMA_ADR-SCHALTER extra!
VORTEX, LASERDRUCKER, SCANNER--
keine Probleme !!!
NETZWERKFÄHIG !!!
LÄUFT UNTER TOS 1.0 bis 2.0
mit allen Rechnern.
SCSI -ADRESSE :
DMA: 0-7
DMA GEPUFFERT IN - OUT
AUTOPARK HARDWAREMÄSSIG
64 kB CACHE_SPEICHER 64 kB SOFTWAREMASSIG, 32 kB HARDWAREMASS
AUTOBOOTFÄHIG VON ALLEN PARTITIONEN, BOOT FREI WAHLBAR - ACCESSORYSTEUERUNG (GEM)
14 PARTITIONEN UNTER TOS, PARTITIONEN EINZELN PARTITIONIERBAR (TOS, DOS..)
KONSTANTGEREGELTER LÜFTER EINZIGARTIGE RETRY - VERIFY - FUNKTION
SUPERLEISE -- NEUESTE LAUFWERKE
EINBAU EINES ZWEITEN CHASSIS - JA !
ALLE PLATTEN HABEN INTERLEAVE 1-1
MHD 50 48 MB (SEAGATE ST 157N) 28 ms, 620kB-s 3.5 zoll 998 DM
MHD 80 85 MB (SEAGATE ST 296N) 28 ms, 550kB-s 5.25 zoll 1.098 DM
MHD 81 84 MB (S -IMPRIMIS 1096 N) 24 ms, 770kB-s 3.5 zoll 1.248 DM
MHD 140 140MB ("-") NUR 15 ms, 1020KB-s, 32kB CACHE 3.5 zoll 1.998 DM
MHD 170 170MB ("-") NUR 15 ms, 1200KB-s, 64kB CACHE 3.5 zoll 2.398 DM
MHD 210 210MB ("-") NUR 15 ms, 1250KB-s, 64kB CACHE 3.5 zoll 2.545 DM
SCSI-BUS: 50 DM
DMA_SCHALTER:
AUFPREIS: 25 DM
neu QUANTUM 1 zoll
17 ms, 1440 kB-s, 64+64 kB CACHE, prog bar
besser und schneller als die alten Quantum-Platten
MHD 52 PRO 52 MB 1.498 DM
MHD 105 PRO 105 MB 1.898 DM
DIE PRO-SERIE WIRD MIT Backup-Programm plus Optimizer ausgeliefert
Alle Platten sofort lieferbar
2 Jahre Garantie !!
Michael Fischer · Computersysteme
Goethestr. 7 · 6101 Fränkisch-Crumbach · Tel. 0 61 64 / 46 01







Tel. 089 / 3089408
Fax 089 / 3085636

Szemere Hard & Software

ATARI Systemfachhändler

Schleißheimerstr.127
D-8000 München 40

PD-Service PD-POOL Mitglied

1040 STF incl. Maus,Mon.und Basic	DM 1099.00
1040 STE incl. Maus,Mon. und Basic	DM 1459.00
Mega ST 1 incl. Maus,Mon. und Basic	DM 1398.00
Mega ST 2 incl. Maus,Mon.und Basic	DM 1799.00
Monitor SM 124	DM 333.00
Monitor SC 1224	DM 558.00
Power Pack -Spiele-	DM 93.00
Megafile 30	DM 889.00
Megafile 60	DM 1239.00
Lynx	DM 358.00
Spiel für Lynx 2 Stück	DM 150.00
Portfolio	DM 498.00
par. Interface Portfolio	DM 89.00
ser. Interface Portf.	DM 159.00

Stacy lieferbar 1/2 auf Anfr.

Die Dauerbrenner

Mega ST 1 + AT-Speed	DM 1897.00
Mega ST 2 + AT-Speed	DM 2279.00
Mega ST 1 +PC-Speed	DM 1796.00
Mega ST 2 +PC-Speed	DM 2159.00

Preis incl.Mon.,Maus und Basic

Zubehör

TOS 1.4	DM 189.00
PC-Speed 1.4	DM 398.00
AT-Speed	DM 529.00
Turbo 16 16 MHz	DM 598.00
Laufwerk 3,5"	DM 249.00
Laufwerk 5,25"	DM 269.00
SUPERCHARGER !MB 1.4	DM 699.00

Speichererweiterungen günstig auf Anfrage !!! auch 12 MB !

Signum II	DM 399.00
1 ST Word Plus 3.15	DM 225.00
Adimens 3.0	DM 298.00
Calamus	DM 699.00
PCB Edit V 2.0	DM 199.00

Disketten 10 Stck nur DM 15.90

Versand

Versandart : UPS
Gebühren : Nachnahme
incl. Porto u. Verpackung DM 16.00
Ausland DM 28.00
Vorkasse

Hendrik Haase Computersysteme präsentiert:

Atari-Computer

Atari 1040 STE	Preissenkung
Atari Mega ST 1	Preissenkung
Atari Mega ST 2	Preissenkung
Atari Mega ST 4	Preissenkung
Atari Megafile 30 Festplatte	750,- DM
Atari Megafile 60 Festplatte	1240,- DM
Wechselplatte 44	Preissenkung
Vortex Datajet 30 Festplatte	1100,- DM
Vortex Datajet 60 Festplatte	1690,- DM
Star Drucker LC 24/10	668,- DM
Epson LQ 550	698,- DM
Fujitsu DL 1100	898,- DM
HP Deskjet 500	1550,- DM
Atari Laserdrucker	Preissenkung
Graustufen Multisync	550,- DM
Mitsubishi Multisync	1298,- DM

Stacy-Computer und TT-Computer
in unterschiedlichen Versionen
TT-Vorführgeräte auf Anfrage

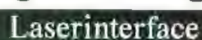
Bestellungen und Informationen bei:

Hendrik Haase Computersysteme
Wiedfeldtstraße 77 • D-4300 Essen 1
Telefon 0201 - 422575 • Fax 0201 - 410421



Die Asse von digital image





Spielen Sie Ihren Trumpf mit dem Laserinterface SLM 804 voll aus. Machen Sie Ihren Atari-Laserdrucker (SLM 605 & 804) mit unserem Steckadapter flusterleise und sparsam. Booten mit abgeschalteten Laserdrucker - kein Problem. Das Steckmodul wird einfach zwischen DMA-Controller und Laserdrucker eingefügt. Test: 7/90 ST - Magazin -› Zitat: » Wir können diesen kleinen "Nervenschoner" jedem empfehlen. «

Laserinterface	DM 98.--
Laserinterface prof.	DM 198.--

HD-Kit

Bringen Sie Ihren Atari mit 1.44 MB Laufwerken auf den Stand der Technik. Kompatibel zu IBM PS/2 und Macintosh II-Systemen. Da das Original dem Rotstift zum Opfer gefallen ist, ist das DI HD-Kit jetzt auch für den Atari TT (alle Modelle) einsetzbar. Komplett anschlußfertig mit umfangreicher Software zum Formatieren, Kopieren, Editieren (Diskettenmonitor) u.s.w. der HD-Disketten. Qualität statt Quantität - Test c't 8/90, TOS 7/90, ST-Magazin 7/90. Kompatibel zu PC + AT-Speed, Vortex ATonce und Supercharger.
HD-Kit 3,5" extern/intern -› 448.--/398.-- DM - nur HD-Board mit Software 198.-- DM - jetzt lieferbar !! --› Original WD 1772 PH 02 nur 49.--

SpeedBridge

Steckadapter zum lötfreien Einbau von PC/AT-Speed oder Vortex ATonce
Für alle Mega ST Rechner geeignet, der Emulator wird einfach auf den Systembus aufgesteckt - fertig. Komplett anschlußfertig nur 89.-- oder mit AT-Speed nur 475.--



Verkaufsburo Rhein-Main
Postfach 1206
D-6096 Raunheim
Tel.(06134) 51706 + FAX 51748



Fachhändleranfragen erwunscht

rationen, die im Fall globaler Prozedurdeklarationen in lokale Deklarationen und Prozedurrumpf unterteilt werden. Letzterer hat dann die einzelnen Anweisungen der Prozedur als Nachfolger usw. Wie solch ein Baum aussehen kann, werden wir nachher noch an einem einfachen Beispiel sehen.

Nach diesen zwei doch sehr krassen Umformungen des Programms kommen wir zu einer subtileren, meist aber auch komlizierteren Umformung. Die semantische Analyse verändert den Strukturbaum meist nicht sehr stark, stattdessen fügt sie den einzelnen Knoten des Baumes weitere Informationen hinzu. Diese Informationen werden als Attribute bezeichnet, und wir können uns einfach vorstellen, daß diese direkt an die Knoten des Baumes gehängt werden, denen sie zugehören. In diesen Attributen wird alle Information gespeichert, die die semantische Analyse berechnet. Dazu gehören Sachen wie die Typen der einzelnen Variablen am Ort ihrer Benutzung, ob ein Malzeichen für eine Ganzzahl- oder eine Fließkommamultiplikation steht, wieviel Platz eine Variable im Speicher braucht, wieviele Parameter eine Prozedur bei einem Prozeduraufruf benötigt und wieviel Platz auf dem Stack für ihre lokalen Daten (ihr Activation Record) gebraucht wird. Die Menge und Art der berechneten Information hängt stark von der übersetzten Programmiersprache ab. Dabei ist eine Sprache wie C recht einfach zu analysieren, da dort viele Überprüfungen, wie der Test auf die korrekte Anzahl an Prozedurparametern oder gar Import und Export zwischen verschiedenen Modulen, gar nicht durchgeführt werden. Schwerer ist es da schon einen Modula-2-Compiler zu schreiben, weil hier sehr viel mehr Attribute berechnet und überprüft werden müssen. Grausam wird es zum Beispiel bei Ada. Hier haben sich schon ganze Scharen von Menschen den Kopf darüber zerbrochen, wie man all die semantischen Regeln einigermaßen geordnet in Attributberechnungen für einen Compiler umwandelt.

Warum die semantische Analyse so kompliziert ist, werden wir in der dritten Folge dieser Serie sehen, die sich fast ausschließlich mit Semantik beschäftigt. Die zwei Phasen nach der semantischen Analyse werden oft weggelassen oder stark eingeschränkt, da sie nicht unbedingt erforderlich sind, beim Compilieren je nach Ausführlichkeit recht viel Zeit kosten können und den Compiler in der Regel ein gutes Stück größer und komplizierter machen. Der Preis, den man für diese Vereinfachung zahlt, ist ein längerer Code und eine geringere Ausführungsgeschwindigkeit des Programms. Die Zwischencode-



erzeugung wandelt den von der semantischen Analyse gelieferten attributierten Strukturbaum, wie ihr Name schon sagt, in einen Zwischencode um. Das Aussehen und die Komplexität des Codes hängt sehr stark von der Quell- und der Zielsprache des Compilers und von der Art der durchzuführenden Codeoptimierungen ab. Meist ist es ein linearer Code, der in vielem schon an Assembler-Code erinnert, aber noch Elemente der Quellsprache besitzt.

Zum Beispiel werden Schleifen normalerweise schon in bedingte Sprungbefehle umgewandelt, aber Array-Zugriffe haben noch ihre alte Form, wie zum Beispiel $a[i]$. Komplexe arithmetische Ausdrücke werden zu sogenanntem 3-Adreß-Code reduziert. Es wird also statt $x := 2 * a + b$ nun $t := 2 * a; x := t + b$ geschrieben. 3-Adreß-Code heißt es deshalb, weil in einer Zuweisung maximal drei verschiedene Variablen und Konstanten vorkommen dürfen. Außer der Zuweisung kann eine 3-Adreß-Code-Anweisung maximal noch einen weiteren Operator enthalten. Der 68000 besitzt zum Beispiel einen 2-Adreß-Code, da in einem Befehl maximal zwei verschiedene Operanden vorkommen. Die Erzeugung des 3-Adreß-Codes aus dem Strukturbaum ist in der Regel eine recht einfache Sache; wie man es genau macht, werden wir in einer späteren Folge sehen.

Viel komplizierter ist hingegen eine gute Codeoptimierung. Hierbei genügt es nicht, die vorhandenen Register des Prozessors geschickt im Code zu verwenden, sondern es muß eine sogenannte Datenflußanalyse durchgeführt werden. Dabei werden alle Wege ermittelt, die der Programmablauf nehmen kann, und es wird errechnet, wie die einzelnen im Programm errechneten Daten auf den einzelnen Wegen Verwendung finden. Dadurch kann der Compiler zum Beispiel Berechnungen, die in einer Schleife gemacht werden, aber bei jedem Durchlauf zum gleichen Ergebnis führen, aus der Schleife herausziehen, oder es kann nie verwendeter Code aus dem Programm eliminiert werden. Wir wollen die Möglichkeiten der Codeoptimierung etwas genauer in der fünften und letzten Folge der Serie kennenlernen. Die Implementierungstechniken werden dabei aber nur gestreift, da die Datenflußanalyse ein recht theoretisches Gebiet ist und eine ausführliche Erklärung den Rahmen dieses Artikels deutlich sprengen würde.

Zu guter Letzt erzeugt der Compiler aus dem optimierten Zwischen- den Zielcode. Dabei werden dann auch die Registerverteilung und die Zuordnung von Speicherplatz an die einzelnen Objekte wie Variablen und Prozeduren vorgenommen. Neben der Verteilung von Speicherplatz (Speicherallokation) muß der Compiler den 3-Adreß-Code nun in den endgültigen Maschinencode umwandeln. Bei einigen größeren Rechnern ist dies sehr einfach, da diese selbst einen 3-Adreß-Code als Maschinencode haben. Komplizierter wird es da schon bei einem Prozessor wie dem 68000, da die 3-Adreß-Code-Anweisungen hier zu 2-Adreß-Code reduziert werden müssen. Zum Beispiel wird aus einer Anweisung wie $x := a + b$ dabei etwa

```
MOVE.W  a, D0
ADD.W   b, D0
MOVE.W  D0, x
```

Außerdem müssen die schon erwähnten Array-Zugriffe codiert werden, indem die Speicherstelle des gewünschten Array-Elements aus dem Index, der Elementgröße und der Speicheradresse des ersten Elements berechnet wird. Somit wird aus $x := a[i]$ zum Beispiel

```
LEA     a, A0
MOVE.W  i, D0
ADD.W   D0, D0
MOVE.W  0(A0; D0.W), D0
MOVE.W  D0, x
```

wenn die Elemente von a zwei Bytes groß sind.

Diese Aufblähung des Codes wird in der Regel mit Hilfe einer Art Makro realisiert, d.h. zu jedem möglichen 3-Adreß-Befehl oder zu einer Gruppe sehr ähnlicher 3-Adreß-Befehle existiert eine Schablone aus 68000-Code, in deren Lücken lediglich noch Sachen wie Adressierungsarten, Konstanten u.ä. eingesetzt werden. Leider können durch dieses sture Aneinanderhängen von Schablonen wieder Ineffizienzen in den Code kommen. Zum Beispiel kann es passieren, daß es für den letzten Befehl einer Schablone und den ersten Befehl einer in einem Programm zufällig darauffolgenden Schablone einen vom Mikroprozessor bereitgestellten Spezialbefehl gibt, der beides in einem und damit schneller erledigt. Solche Optimierungen können zum Schluß noch von einem sogenannten Peep-Hole-Optimierer (wörtlich übersetzt Gucklochoptimierer) durchgeführt werden. Dieser schaut sich den erzeugten Maschinencode zwar von vorne bis hinten an, allerdings betrachtet er dabei immer nur zwei bis drei Befehle

gleichzeitig. Sobald diese kurze Befehlssequenz einem von mehreren gespeicherten Mustern entspricht, also zum Beispiel aus der vorhin erwähnten Folge von zwei Befehlen am Berührungspunkt zweier Schablonen besteht, setzt er dafür einfach den entsprechenden Spezialbefehl ein. Dabei dürfen natürlich nur Befehlskombinationen berücksichtigt werden, die der Compiler überhaupt erzeugen kann und die häufig vorkommen.

Wir werden die Codeerzeugung ausführlich in der fünften Folge besprechen und dabei auch sehen, wie man alternativ zur Methode mit dem Zwischencode direkt aus dem attributierten Strukturbaum einen ganz anständigen Code erzeugen kann.

Fließbandarbeit

Bei der beschriebenen Methode, einen Compiler zu bauen, ist es natürlich nicht nötig, daß der gesamte Quelltext von einer Phase komplett verarbeitet und dann an die nächste weitergereicht wird. Der Scanner kann zum Beispiel ein einzelnes Token erkennen und sofort an den Parser weitergeben. Dieser verarbeitet das Token und kehrt dann zum Scanner zurück, um diesem die Möglichkeit zu geben, das nächste Token zu erkennen.

Auf diese Art können alle Phasen (1-Pass-Compiler) oder einzelne Gruppen gemeinsam ineinander verzahnt arbeiten. Man kann sich das wie Fließbandarbeit vorstellen: Der erste Arbeiter nimmt ein paar Einzelteile, baut sie zusammen und legt das Ergebnis auf das Fließband, wo es zum zweiten Arbeiter transportiert wird, der noch ein weiteres Teil anschraubt. Ist er damit fertig legt er sein Produkt wieder auf das Fließband zurück und wartet auf das nächste Teil, das vom ersten Arbeiter zu ihm kommt, usw. Diese Methode hat den Vorteil, daß die Zwischenformen des Quelltextes gar nicht oder nur selten komplett im Speicher vorliegen, was natürlich Speicherplatz oder, beim Ablegen auf Disk, Zeit spart. Der Nachteil ist, daß die Konstruktion des Compilers schwieriger wird, da alle Phasen sehr genau aufeinander abgestimmt sein müssen. Außerdem gibt es Situationen, in denen es nicht möglich ist, alle Phasen gleichzeitig ablaufen zu lassen. Aber das werden wir später noch genauer sehen.

1-Pass-Compiler machen zum Beispiel alles vom Scanner bis zur Codeerzeugung in einem Rutsch. Ein 2-Pass-Compiler wandelt die Eingabe einmal komplett in eine der Zwischenformen um, zum Beispiel in den Zwischencode, und übersetzt von dort aus in den eigentlichen Zielcode. Dabei werden zunächst die Arbeitsschritte

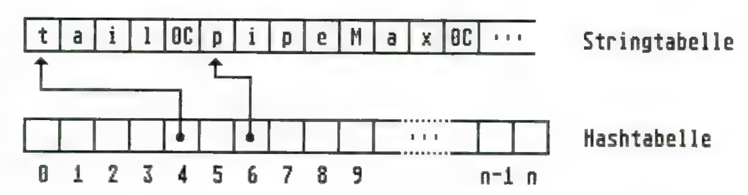


Abb. 2: Beispiel einer Bezeichnertabelle

vom Scannen bis zur Zwischencodeerzeugung in einem Zug zusammen, und der so entstandene Zwischencode wird mit oder ohne Codeoptimierer von der Codeerzeugung in Maschinencode umgewandelt.

Egal, welche Technik genau verwendet wird, wichtig ist immer, daß der Compiler von vornherein gut gegliedert und die Aufgabe der einzelnen Komponenten und ihrer Schnittstellen klar umrissen ist. Beinahe unnötig zu erwähnen, daß eine Programmiersprache wie Modula-2, die eine solche Modularisierung von Anfang an unterstützt, dabei eine große Hilfe ist.

Achtung: Lawinengefahr!

Nachdem wir die einzelnen Transformationsphasen eines Compilers kennengelernt haben, beschäftigen wir uns mit den Komponenten, die von allen Phasen gemeinsam verwendet werden. Die wichtigsten davon sind die globalen Tabellen, also die Bezeichner-, Definitions- und Konstantentabelle, sowie die Fehlerbehandlung des Compilers. Sie ist eine recht wichtige Komponente, da eine gute Fehlerbehandlung die Arbeit des Programmierers, der mit dem Compiler arbeitet, stark erleichtern kann. Wichtig ist dabei, daß der Compiler möglichst genaue und gut verständliche Fehlermeldungen liefert und nach Möglichkeit in der Lage ist, eine Fehlerliste statt eines einzigen Fehlers zu liefern. Allerdings darf diese Liste keine sogenannten Fehlerlawinen enthalten, und sie sollte möglichst in einer Form ausgegeben werden, die ein Editor nutzen kann. Was ist nun eine Fehlerlawine? Schreibt man in einer Variablendeklaration zum Beispiel den Typ der Variablen falsch, und es wird deshalb ein Fehler ausgelöst, sollte der Compiler bei allen späteren Vorkommen dieser nicht korrekt deklarierten Variablen alle diejenigen Fehlermeldungen unterdrücken, die mit der falschen Deklaration zu tun haben. Wie man nach einem Fehler am geschicktesten weiterarbeitet, werden wir in den Folgen über die syn-

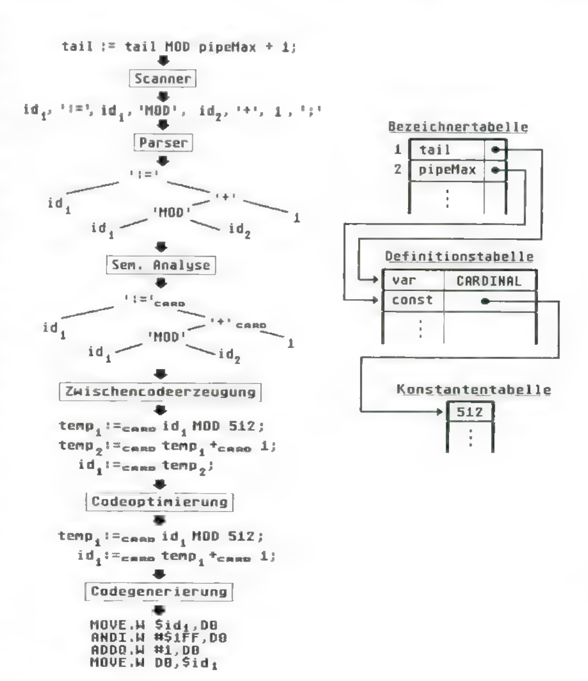


Abb. 3: Transformation eines arithmetischen Ausdrucks

taktische und die semantische Analyse sehen.

Globale Tabellen

Die Funktion der Bezeichner- und der Konstantentabelle haben wir weiter oben schon angerissen. Implementiert werden die beiden Tabellen meist durch ein Stück linearen Speichers (also meist ein Array), in das alle vom Scanner gefundenen Bezeichner bzw. Konstanten einfach nacheinander, durch ein Trennzeichen (z.B. 0C) getrennt, eingetragen werden. Im Rest des Compilers wird statt den Bezeichnern bzw. Konstanten dann der Index des ersten Array-Elements, das von dem jeweilen Objekt belegt wird, verwendet. In Abb. 2 wird eine derart aufgebaute Tabelle gezeigt, in die die Bezeichner *tail* und *pipeMax* eingetragen wurden.

Bei der Bezeichnertabelle ist es im Gegensatz zur Konstantentabelle noch nötig, daß alle Bezeichner mit gleichem Namen auch durch die gleiche Nummer repräsentiert werden. Dies läßt sich am besten durch die Verwendung einer Hash-Tabelle realisieren, die der soeben beschriebenen Tabelle vorangestellt wird (siehe Abb. 3). Eine Hash-Tabelle beschleunigt dabei das Auffinden gleicher Bezeichner. Die Funktionsweise einer Hash-Tabelle zu erläutern, würde hier zu weit führen: wer interessiert ist, kann eine kurze Beschreibung in einer älteren ST-Ecke ([1]) finden. Ausführlicher steht es zum Beispiel in [2]. In dem Beispiel aus der Abbildung werden *tail* durch die Zahl 4 und *pipeMax* durch die Zahl 6 repräsentiert.

Definitionssache

Zusätzlich zum Namen und der für einen Bezeichner verwendeten Nummer, muß der Compiler natürlich auch noch die Bedeutung jedes im Programm vorkommenden Bezeichners speichern. Erschwert wird das durch die in vielen Sprachen vorhandene Möglichkeit, globale Deklarationen lokal durch andere Deklarationen des gleichen Namens zu überdecken. Das Überdecken kann zum Beispiel durch Stacks implementiert werden. Dabei ist immer die oberste Deklaration gültig. Sobald man eine Ebene tiefer gehen möchte, wird einfach das oberste Element vom Stack heruntergenommen. Der genaue Aufbau einer Definitionstabelle ist in der Regel recht trickreich und hängt stark von der zu übersetzenden Programmiersprache ab. Deshalb gehen wir hierauf nicht detaillierter ein. Einen recht guten, für viele Programmiersprachen verwendbaren Vorschlag zu Definitionstabellen kann man in [3] nachlesen.



Ein Beispiel

Nachdem wir uns durch den kompletten Aufbau eines Compilers gekämpft haben, gibt es ein anschauliches Beispiel zur Belohnung. In Abb. 3 sehen wir noch einmal die Komponenten eines Compilers, diesmal aber mit einem Stück Quelltext und den einzelnen Zwischenformen, die dieses Programmfragment durchläuft, bis es zum Schluß in 68000-Code umgewandelt wird.

Am Anfang steht die Zuweisung *tail := tail MOD pipeMax + 1*. Der Scanner macht daraus eine Folge von Tokens, dabei werden die Tokens ':=', '+', ';' und *MOD* allerdings nicht mehr durch ihre Zeichenfolgen dargestellt - in der Abbildung geschieht dies nur der Übersicht halber. Wichtig ist, daß die Bezeichner *tail* und *pipeMax* nur noch durch das allgemeine Bezeichnertoken *id* und ihre Referenznummer, hier 1 und 2, repräsentiert werden.

Diese lineare Folge von Tokens wird vom Parser in einen Strukturbaum umgeformt. Die Wurzel dieses Baumes ist die Zuweisung ':='. Ihre Nachfolger sind zur Linken das Ziel der Zuweisungsoperation (in unserem Beispiel der Bezeichner Nummer 1) und zur Rechten der Ausdruck, der den zuzuweisenden Wert errechnet. Dieser Ausdruck besteht aus den beiden Operatoren '+' und *MOD* sowie ihren Operanden Bezeichner 1, 2 und der Konstanten '1', die die Blätter des Baumes bilden. Warum sieht der Baum so aus und nicht anders? Die Grundregel ist ganz einfach: Will man den im Baum gespeicherten Code ausführen, arbeitet man sich von den Blättern zur Wurzel nach oben durch, d.h. zuerst werden die Blätter ausgewertet, was sehr einfach ist, da man nur den Wert des Bezeichners bzw. der Konstante einzusetzen braucht. Danach wird die Operation *MOD* ausgeführt, also der Rest der Ganzzahldivision zwischen den Werten der Bezeichner Nummer 1 und Nummer 2 ermittelt. Dann, wieder eine Stufe nach oben gehend, wird zum Ergebnis der *MOD*-Operation eine 1 addiert. Schlußendlich wird das Ergebnis an die Speicherstelle des Bezeichners Nummer 1 geschrieben. Die Zuweisungsoperation ':=' behandelt ihre linke und rechte Seite also verschieden, da von der rechten Seite der Wert und von der linken die Speicherstelle ermittelt wird. Das rührt daher, daß man in einer imperativen Sprache - und nur die sollen uns interessieren - bei einer Variablen immer zwischen ihrem Wert und ihrem Speicherplatz unterscheiden muß, obwohl beides durch denselben Namen repräsentiert wird. Die C-Fans bezeichnen die verschiedenen Bedeutungen ein und desselben Namens zum Beispiel als *l-value* und *r-value*, wobei sich *l* bzw. *r* von left bzw. right hand side (of an assignment) ableitet.

Überlast

Nachdem der Parser den Strukturbaum ermittelt hat, wird dieser an die semantische Analyse weitergeleitet. Diese findet jetzt anhand der Information, die in der Definitionstabelle über die Bezeichner Nummer 1 und 2 eingetragen ist, heraus, daß sowohl die Addition als auch die Zuweisung Werte vom Typ CARDINAL behandeln müssen. Diese Analyse muß gemacht werden, da in den meisten Programmiersprachen einige Operatoren überladen sind, d.h. daß man denselben Operator für verschiedene Funktionen benutzt. In unserem Fall kann das Pluszeichen '+' sowohl zwei CARDINAL als auch zwei INTEGER oder zwei REAL-Zahlen addieren. Es hat somit nicht nur eine, sondern mehrere Bedeutungen, ist also mit Funktionen überladen. Noch schwieriger wird eine solche Analyse übrigens, falls auch der Benutzer überladene Operatoren deklarieren kann, wie das in Ada der Fall ist.

Offen ist noch, woher die Einträge in die Definitionstabelle stammen, die hier von der semantischen Analyse verwendet werden. Ganz einfach: Irgendwann mußten ja die Variable und die Konstante deklariert werden, zum Beispiel mit *CONST pipeMax = 512;* und *VAR tail: CARDINAL;* (für die C-Programmierer: *CARDINAL* entspricht *unsigned*). Auch dieser Teil des Programms wurde natürlich von der semantischen Analyse bearbeitet, und dabei wurden auch die entsprechenden Eintragungen in die Definitionstabelle gemacht. Ein Problem taucht natürlich auf, wenn die Deklaration von *pipeMax* und *tail* im Programmtext hinter ihrer Benutzung kommt, was zum Beispiel in Modula-2 unter gewissen Umständen möglich ist. Vernünftig kann eine solche Situation dann nur von einem 2-Pass-Compiler gelöst werden, der im ersten Pass alle Deklarationen abarbeitet und erst beim zweiten Durchlauf die Verwendung der einzelnen Objekte überprüft. Von 1-Pass-Compilern werden solche Deklarationen normalerweise verboten, was allerdings auch keine

wesentliche Einschränkung ist. Abschließend soll zur semantischen Analyse noch erwähnt werden, daß eine Fehlermeldung ausgelöst wird, sobald einer der Operanden nicht vom Typ *CARDINAL* ist.

Der Code

Nachdem die semantische Analyse mit dem Strukturbaum fertig ist, kann jetzt der Zwischencode erzeugt werden. Dabei wird, wie schon erwähnt, von unten nach oben zur Wurzel hin gearbeitet. In unserem Beispiel wird also zuerst die MOD-Operation in Zwischencode übersetzt, dann die Addition und zum Schluß die Zuweisung. Bei dieser Umwandlung werden vom Zwischencodegenerator Hilfsvariablen eingeführt, die hier mit *temp* bezeichnet sind. Sie sind nötig, da in dem benutzten 3-Adreß-Code pro Zeile außer der Zuweisung nur noch eine weitere Operation erlaubt ist. Der 3-Adreß-Code wird dann vom Codeoptimierer weiterverarbeitet. In unserem kurzen Beispiel passiert dabei nicht viel. Es werden lediglich die letzten beiden Anweisungen zu einer verschmolzen, wobei eine der beiden Hilfsvariablen eliminiert wird.

Der so optimierte Code wird nun vom Codegenerator in 68000-Assembler-Code umgesetzt. Dabei wird für die Hilfsvariable in unserem Beispiel das Register *D0* verwendet, und der Codegenerator ist so schlau, daß er statt einer normalen *ADD*-Operation ein *ADDQ* verwendet, und erkennt, daß eine Modulooperation mit einer 2er-Potenz durch ein *AND* ersetzt werden kann. Diese beiden Optimierungen können zum Beispiel von der oben erwähnte Peep-Hole-Optimierung durchgeführt werden. Der Codegenerator kann den Code ansonsten mit dem Schablonenverfahren erzeugen, das wir weiter oben besprochen haben.

Und wie geht's weiter?

Nachdem wir diesmal gesehen haben, was ein Compiler können muß und wie er das im Groben bewerkstelligt, werden wir ab der nächsten Folge etwas genauer auf die einzelnen Probleme eingehen und einige Standardlösungen kennenlernen.

Manuel Chakravarty

Literatur:

[1] Chakravarty/Keller/Höhn: „Verpackungskünstler", ST-Computer 5/90

[2] Aho/Sethi/Ullman: „Compilers: Principles, Techniques and Tools", Addison-Wesley

[3] Waite/Goos: „Compiler Construction", Springer

Public-Domain Software

Hier stellen wir Ihnen unsere neuesten ACTION - Pakete vor. Die Pakete werden alle drei Monate erneuert und wieder mit aktuellster PD-Soft bespielt!

Jedes ACTION - Paket besteht aus 3 doppelseitigen PD-Disketten.

Jedes Paket kostet nur DM 10,--

Ausgabe: 12/90 bis (einschl.) 2/91

GAS.-01/ *Monochromspiele - Action* (3 Disks, DM 10,--) **Darkness is for ever:** ein Adventure, daß Sie viele Stunden in seinen Bann ziehen wir **Rennhund:** sportliches Managerspiel für 1 bis 8 Teilnehmer. **Quizmaster 0.5:** nettes Ratespiel für 1 bis 2 Spieler **Knack den Tresor:** sehr aufwendiges Spiel, bei dem Sie 40 alte Bilder erstehen mussen **Chicago:** Glucksspiel **Play Me:** sehr schone Tetris-Variante. **Hinem:** Memoryspiel mit 198 Karten **Cave Miner II:** super Boulder-Dash-Variante. **Fly A Jet:** Flugsimualtor einer Boing 727, mit vielen Funktionen **Segelflug:** Segelflug-Simulator **Memory:** 'normale Memory-Variante □

GAF.-02/ *Farbspiele - Action* (3 Disks, DM 10,--) **Persifal:** spannendes Abenteuerspiel **Megamix 1.2:** Memoryspiel **TriDo 1.5:** super Geschicklichkeitsspiel **Xenotron 2.3:** schnelles Weltraum-Ballerspiel. **Crisslefridge:** geniales Boulder-Dash Spiel mit super Grafik. **Two for Two:** Panzerspiel, das eine schnelle Reaktion und gute Nerven voraussetzt **Double Bounce:** Tennisspiel fur 2 Sportler **Gilbert:** Memo-Variante. **Deathstar:** Kampf gegen Monster und vieles mehr **Damonoid:** abwechslungsreiches Ballerspiel mit digit. Stimmen **Shymer:** Adventure □

ANW.-01/ *Anwender - Action* (3 Disks, DM 10,--) **Beropress 1.0:** Druckt Poster, Grußkarten etc. Verarbeitet sogar Signum-Zeichensatze **Formular 1.1:** Schreibprog zum Ausfüllen von Formularen **Stundenplan. Bildklau:** ladt Bilder ein und verarbeitet diese in Data-Zeilen für GfA-Basic **Disk Manager:** archiviert Disketteninhalte. **Houdini:** Tool fur Signum. **Getfish - Tab:** verwaltet bei Sportligen **Adress 5.0:** Adreßverwaltung.□

MUG.-01/ *Musik und Grafik - Action* (3 Disks, DM 10,--) **Van Gogh III:** eines der besten Mal/Zeichenprogramme. **Click & Beep:** belegt die Tastatur mit digit Sounds **Butterfly Artist:** ausgefa enes Malprogramm **Freedrum 2.5:** Sound/Trommelmaschine mit digit Effekten. **ST-Design:** leistungsfahiges Zeichenprog.□

CLP.-01/ *Clip Art - Action* (3 Disks, DM 10,--) die besten Clips im PAC-Format (Stad) □

SIG.-01/ *Signum Zeichensätze - Action* (3 Disks, DM 10,--) ca 30 verschiedene Signum (Script) Zeichensatze (fur 9 und 24 Nadeldrucker), sowie verschiedene Tools □

Fordern Sie auch bitte unseren Gesamtkatalog (gedruckt) für DM 3,-- an.

Ach: Bei uns kostet jede PD-Disk nur DM 4,--

COMPUTER-SERVICE KOHLER

Don-Carlos-Str. 33B, D-7000 Stuttgart 80

Telefon: 07 11 / 6 78 73 92 (24h-Bestellservice)





ATARI ST Registrierkasse
ST+Drucker - Beleg Schmal-o.Normaldruck,auch fur Beleg-Drucker - Protokoll auf Disk, ausdruckbar - Leistungen/Artikel von Disk o.Hand - Firmendaten - Werbeslogan - Kassenstand - Kassierermarke DM 146.-

GESCHÄFT ATARI ST
Editor f. Formular-, Adressen-, Artikel- +Dienstleistungsdateien - Angebot/Voranschlag., Auftr.Bestatigung, Auftrag/Bestellung, Rechnung, Lieferschein, Mahnung - Eingabe Hand o.Datei - Durchrechnung u. Menge Preis, Aufschlag/Rabatt, MWSteuer, Skonto usw. - Verpackung-/Versand-Angaben - Editor für Textfeld - Kein Datenverbund mit Lager-/Finanz-Buchhaltung DM 196.-

ATARI ST Inventur, Fibu-gerecht
Kontinuierl. Lager-Bestandsverwaltung m.Bild-Moment u./o.Listenauswertung - Tages- bis Jahres-Neuinventur d. Streichen/Ändern/Hinzufugen - Gruppenauszuge nach Code - Bis 3000 Positionen/Datei DM 116.-

Provisionsabrechnung ATARI ST
Editor f.Vertreter-, Kunden- u.Firmen dateien - Eingabe von Hand/Datei - Prov.Satz -99.99% Storno-Spesen - Endbetrag m.o.MWSteuer - Ausdruck DM 116.-

ATARI ST TYPIST
Der ST+Drucker als Elektronik-Schreibmaschine - Ausdruck zeilenweise - 15 Zeilen Bildschirm-Display - Korrektur - Je nach Drucker bis zu 30 Schriften - Ablage auf Disk - Kopie-Ausdruck - Super ! DM 86.-

Etikettendruck ATARI ST
Druckt Auflagen von 40 gangigen Lochrand-Haftetiketten-Formaten - Texteingabe in jeweils passende Bildschirmmaske - Ablage auf Disk fur jederzeitige Neuauflage - Schriftenwahl n.Drucker-Handbuch DM 89.-

ATARI ST BACKGAMMON
Das Strategie+Gluck-Spiel - Bestechende Grafik - In Schwarz/Weiß und Farbe - Ausf. Anleitung DM 58.-



Welche Firmen / Privatleute / Softwareleute / Hardwareleute haben Interesse, einen neuen Hardware-/Software-Standard auf Basis der 68xxx Familie zu setzen?

Die Zeit dazu ist günstig. MS-DOS mit seinem Speicher von 640 KByte ist für größere Anwendungen zu klein. Unix ist zu teuer.

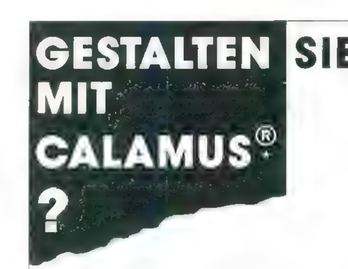
Deshalb haben wir die Chance, einen neuen offenen Hardware-/Software-Standard zu setzen. Das Wesentliche an diesem Standard ist die Möglichkeit, für jede Firma eine kombinierte Hardware-/Software-Lizenz zu bekommen. Angefangen vom 68 008, 68 000, 68 010, 68 070 bis zum 68 020, 68 030, 68 040 und Peripherie-Bausteinen der 68xxx Familie.

Wir suchen Finanziers, Hardwareentwickler und Softwareentwickler.

Informationen und Anfragen bitte nur schriftlich an

LB Computer
PF 1253 • 8425 Neustadt/ Donau-Geibenstetten

Wir rufen zurück.



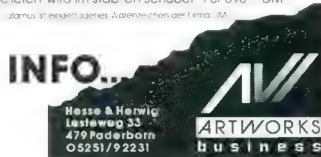
BUSINESS!

Briefbögen
Visitenkarten
Aufkleber
Nutzeneinbindungen
jede Menge Formulare
mit Passermarken
Falz-u. Positionsmarken
und vieles mehr

Mit dem neuen Business-Gestaltungspaket von ARTWORKS COLLECTION steht Ihnen jetzt dieses und noch vieles mehr copyright frei und fertig gestaltet zur freien Verfügung

Gebrauchsfertig angelegt in über 80 CDK-Dokumenten. Dazu im umfangreichen Handbuch mit vielen zusätzlichen Informationen zu Gestaltung und Typografie, Layout und Druckvorlagenerstellung mehrfarbig abgebildet

Geliefert wird im stabilen Schuber Für 398 – DM



Datenstrukturen & Algorithmen

in Omikron.BASIC und Modula 2

Datentypen und Datenstrukturen sind ein wesentlicher Baustein der höheren Programmierung. Wir stellen Ihnen Konzepte zur Realisierung von Datenstrukturen in Omikron.BASIC vor. Aufbauend auf der Cursorimplementierung von Zeigern erläutern und programmieren wir lineare Listen.

Datenstrukturen bilden neben der Beherrschung der Programmiersprache die Basis für gute Programme. Aus der Wahl von geeigneten Datenstrukturen resultieren Laufzeit und Speicherbedarf. Dieser Artikel kann keine vollständige Abhandlung zum Thema Datenstrukturen geben. Er bildet die Grundlage für das Selbststudium mit Lehrbüchern. Die Literatur orientiert sich hauptsächlich an Pascal-ähnlichen Programmiersprachen wie zum Beispiel Modula 2. Folgendes Vorgehen ist daher sinnvoll:

Wir beschreiben alle Sachverhalte exemplarisch in Modula 2. Die Sprachkonzepte erläutern wir soweit, daß Sie Modula 2-Programme lesen und verstehen lernen. Anschließend implementieren wir das Gelernte Schritt für Schritt in Omikron.BASIC.

Wir geben Ihnen das Handwerkszeug, um Datenstrukturen und Algorithmen aus Standardliteratur in Omikron.BASIC-Programme umzusetzen. In diesem Artikel betrachten wir exemplarisch lineare Listen. Nach dem Studium sind Sie in der Lage, sich in andere wichtige Kapitel einzuarbeiten. Beispielsweise programmieren Sie Bäume in Omikron.BASIC - basierend auf Literatur, die die Zusammenhänge in Modula 2 beschreibt.

Drei Begriffe bilden einen Teil der theoretischen Grundlage dieses Artikels: Datentyp, Datenstruktur und abstrakter Datentyp.

Datentyp

Ein Datentyp bezeichnet den Wertebereich eines Objekts. Ein Objekt kann eine Konstante, Variable oder der Rückgabewert einer Funktion sein. Auch Elemente einer linearen Liste sind derartige Objekte. Der Wertebereich ist die Menge von Werten, die ein Objekt annehmen kann.

Omikron.BASIC kennt die elementaren Standard-Datentypen Long-Integer, Integer-Word, Integer-Byte, Boolean, Single-Float, Double-Float und String. Seite 21 des Handbuchs listet diese auf. Betrachten wir beispielsweise 'Long-Integer': Der Wertebereich sind die ganzen Zahlen von -2^31-1 (= -2147483648) bis +2^31 (= +2147483647). Eine Long-Integervariable kennzeichnet das Omikron.BASIC mit dem Postfix *%L*.

Einen sehr wichtigen Datentyp kennt das Omikron.BASIC nicht: den Pointer. Ein Pointer ist ein Zeiger auf den Wert einer Variablen von beliebigem, aber festem Datentyp.

Datenstruktur Feld

Die Zusammenfassung von Werten eines oder verschiedener Datentypen bezeichnet man als *Datenstruktur*. Omikron.-BASIC unterstützt die elementare Datenstruktur *Feld* (Array). Das Feld ist eine homogene Datenstruktur, denn alle Elemente (Komponenten) haben den gleichen Datentyp: den sogenannten Grundtyp. In Omikron.BASIC muß der Grundtyp ein Standard-Datentyp sein - in Modula 2 ist ein selbstdefinierter Datentyp zulässig.

Felder müssen Sie in Omikron.BASIC vor der ersten Benutzung im Programm mit dem *DIM*-Befehl dimensionieren. Sie geben damit an, wieviele Werte des Grundtyps Sie maximal abspeichern. *DIM Text$(100)* fordert beispielsweise vom Omikron.BASIC ein Feld namens *Text* mit 101 Elementen des Grundtyps String. Auf die einzelnen Strings des Feldes greifen Sie mit Indizes zwischen 0 und 100 zu. *Text$(0)* ist die erste Zeichenkette, *Text$(9)* die zehnte und *Text$(100)* die letzte (101.). Man bezeichnet die Datenstruktur Feld auch als Random-Access-Struktur (random access = wahlfreier Zugriff), da Sie auf jedes Feldelement direkt über einen Index zugreifen können.

Die Standard-Datentypen Integer-Byte und Boolean (Flag) sind ausschließlich als Grundtypen für Felder konzipiert. Sie dürfen ein Feld von Integer-Byte-Elementen benutzen, aber keine einzelne Integer-Byte-Variable.

Näheres zum Thema Felder finden Sie im Handbuch ab Seite 66. Wichtig sind die Aussagen zur Redimensionierung.

Datenstruktur Record

Als *Record* bezeichnet man eine Datenstruktur, deren Komponenten verschiedene Bedeutungen haben. Ein Record ist meist aus verschiedenen Datentypen zusammengesetzt. Angenommen, Sie möchten folgende Personendaten speichern: Name, Vorname, Alter und Telefonnummer. In Modula 2 definieren Sie eine Datenstruktur

als Datentyp, die genau diese Daten aufnimmt.

```
TYPE Person = RECORD
    Name      : String;
    Vorname   : String;
    Alter     : INTEGER;
    Telnr     : LONGINT
END;
```

Der so definierte Record-Datentyp heißt *Person*. Sie dürfen ihn wie jeden anderen Datentyp verwenden. Person enthält vier Komponenten. Die ersten beiden sind Strings, die beiden letzten sind ganze Zahlen. Jede Komponente besteht aus Datentyp und Name. Beispielsweise hat die zweite Komponente den Namen *Vorname* und den Datentyp *String*.

Im allgemeinen Sprachgebrauch benutzt man die Begriffe *Datentyp* und *Datenstruktur* oft synonym. Man spricht von der Datenstruktur *Person*, obwohl *Person* genaugenommen ein selbstdefinierter Datentyp ist.

In Modula 2 müssen Sie jede Variable im Programm- oder Prozedurkopf deklarieren. Sie dürfen Variablen nicht einfach benutzen, wie dies in Omikron.BASIC üblich ist. Mit *VAR <varname> : <datentyp>* fordern Sie vom Compiler eine Variable mit dem Namen <varname> vom Datentyp <datentyp>.

Typdefinition und Variablendeklaration sind streng zu trennen. Die Typdefinition legt den Wertebereich fest, den alle Variablen dieses Typs haben. Dies ist reine Verwaltungsarbeit. Erst die Variablendeklaration besorgt Speicherplatz für einen Wert des angegebenen Datentyps.

VAR P1, P2 : Person; deklariert die beiden Variablen *P1* und *P2* vom Datentyp *Person*. *P1* und *P2* nehmen jeweils einen Personen-Datensatz auf - sie sind Variablen, die Datenstrukturen speichern. In Modula 2 weisen wir den Variablen folgendermaßen Werte zu:

```
P1.Name:="Musterfrau";
P1.Vorname:="Brigitte";
P1.Alter:=28;
P1.Telnr:=3116;
P2.Name:="Mustermann";
P2.Vorname:="Klaus";
P2.Alter:=22;
P2.Telnr:=58375;
```

':=' bedeutet in Modula 2 eine Zuweisung. Das Gleichheitszeichen '=' verwendet man bei Vergleichen.

Jede Variable enthält vier Werte. Die vier Komponenten sind zu einer Einheit verschmolzen. Sie greifen auf den Wert einer Komponente zu, indem Sie zuerst den Variablennamen angeben, dann einen Punkt und zum Schluß den Komponentennamen.

Modula 2 gestattet Felder von Datenstrukturen: *VAR Personendaten : ARRAY [0..99] OF Person;* deklariert ein Feld namens *Personendaten*, das die im Datentyp *Person* angegebenen Daten von 100 Personen speichert. Genaugenommen ist *Personendaten* eine Datenstruktur (Feld), deren Elemente wiederum Datenstrukturen (Records) sind.

Wir behalten unsere Sprechweise bei: *Komponenten* bezeichnen Teile einer Datenstruktur. *Elemente* nennen wir die Objekte in einem Feld.

Implementierung von Records

Omikron.BASIC kennt keine Mechanismen, um Record-Typen zu definieren und zu bearbeiten. Wir behelfen uns, indem wir sinnvolle Variablennamen vergeben. Wie? Wir wählen als Variablennamen die Schreibweise des Zugriffsmechanismus' aus Modula 2. Dabei ersetzen wir den Punkt durch einen Unterstrich: In Modula 2 greifen wir mit *P1.Vorname* auf die Komponente *Vorname* der Variablen *P1* zu. In Omikron.BASIC speichern wir diesen Wert in der Variablen *P1_Vorname$*. Diesen Variablennamen interpretieren wir als Komponente *Vorname* der ersten Person. In Omikron.BASIC müssen Sie den Datentyp der Variablen als Postfix angeben. Unsere obigen Variablenzuweisungen lauten in Omikron.BASIC demnach:

```
P1_Name$="Musterfrau"
P1_Vorname$="Brigitte"
P1_Alter%=28
P1_Telnr%L=3116
P2_Name$="Mustermann"
P2_Vorname$="Klaus"
P2_Alter%=22
P2_Telnr%L=58375
```

Modula 2 speichert diese acht Werte in zwei, Omikron.BASIC in acht Variablen. Wenn wir in diesem Artikel von einem Record in Omikron.BASIC sprechen, meinen wir die Variablen für alle Komponenten zusammengenommen. Also der Omikron.BASIC-Record *P2* bezeichnet die vier Variablen *P2_Name$*, *P2_Vorname$*, *P2_Alter%* und *P2_Telnr%L*.

Modula 2 weist bei *P1:=P2* allen Komponenten von *P1* den Wert der entsprechenden Komponenten von *P2* zu. In Omikron.BASIC müssen wir vier Einzelzuweisungen schreiben:

```
P1_Name$=P2_Name$
P1_Vorname$=P2_Vorname$
P1_Alter%=P2_Alter%
P1_Telnr%L=P2_Telnr%L
```

Wichtig ist, daß alle Komponenten den neuen Wert zugewiesen bekommen. Vergessen Sie eine, kann es zu Datensalat kommen.

Auch Felder von Record-Typen 'simulieren' wir durch geschickte Vergabe von Variablennamen. Das obige Array *Personendaten* zur Aufnahme von 100 Records dimensionieren wir in Omikron.BASIC wie folgt:

```
DIM Personendaten_Name$(99)
DIM Personendaten_Vorname$(99)
DIM Personendaten_Alter%(99)
DIM Personendaten_Telnr%L(99)
```

Diese vier Felder bezeichnen wir als Feld *Personendaten*.

Das Problem der vergessenen Komponenten bei Zuweisungen umgehen wir durch Prozeduren für die elementaren Operationen auf den Record. Wir schreiben uns beispielsweise folgende Prozeduren:

```
Speichere_Personendaten(Index%,Name$,
                Vorname$,Alter%,Telnr%L)
Lies_Personendaten(Index%,R Name$,R
                Vorname$,R Alter%,R Telnr%L)
Kopiere_Personendaten(Quell_Index%,
                Ziel_Index%)
Vertausche_Personendaten(Index1%,Index2%)
```

Die Index-Variablen sind die Feldindizes der Elemente, auf die wir zugreifen. Den Programmtext der Prozeduren zeigt Listing 1.

Anstelle der vier Anweisungen

```
Personendaten_Name$(5)="Müller"
Personendaten_Vorname$(5)="Heinz"
Personendaten_Alter%(5)=32
Personendaten_Telnr%L(5)=4711
```

schreiben wir

```
Speichere_Personendaten(5,"Müller",
                "Heinz",32,4711)
```

Kopieren ist noch einfacher:

```
Kopiere_Personendaten(5,3)
```

kopiert den Record *Personendaten(5)* nach *Personendaten(3)*.

Ein Programm ist durch diese Prozeduren wesentlich übersichtlicher und fehlerunanfälliger. Außerdem sparen wir Programmspeicherplatz. Bei einem Fehler wissen wir, daß er entweder in der Prozedur oder beim Prozeduraufruf liegt. Wir lokalisieren Fehler leichter, da sämtliche Datenzugriffe auf wenige Prozeduren beschränkt sind.

Voraussetzung für diese Vorteile ist, daß wir für Zugriffe auf das Record-Feld ausschließlich diese Prozeduren benutzen. Selbst wenn wir nur den Vornamen einer Person ändern. Wir schreiben *Speichere_Personendaten(5,"","Peter",0,0)* anstelle von *Personendaten_Vorname$(5) ="Peter"*.
Die Prozeduren abstrahieren von der Realisierung des obigen Datentyps Person.

Bei ausschließlicher Benutzung der Prozeduren brauchen sie nicht zu wissen, wie die Daten im Computer gespeichert sind. Die Felder *Personendaten_xxx()* sind außerhalb der Prozeduren unwichtig - genau das bedeutet *Abstraktion*: Wichtiges von Unwichtigem zu trennen.

Abstrakter Datentyp (ADT)

Die Informatik spricht in diesem Zusammenhang von einem *abstrakten Datentyp* (ADT). Ein abstrakter Datentyp definiert einen Datentypen und eine Menge von Operationen auf diesen Typ, in unserem Beispiel den Datentyp *Person* mit den Operationen Speichern, Lesen, Kopieren und Vertauschen von Personendaten. Wichtig: Diese Definition ist unabhängig von einer Rechnerimplementierung oder einer bestimmten Programmiersprache. Wir hätten den Datentyp *Person* auch anders als in den vier Feldern *Personendaten_xxx()* speichern können.

Um Sie nicht am Anfang mit dem Begriff *abstrakter Datentyp* zu vergraulen, haben wir zuerst die Implementierung und dann die Theorie vorgestellt. In der Praxis geht man genau umgekehrt vor: Man definiert zuerst einen abstrakten Datentypen mit den zugehörigen Operationen. Anschließend überlegt man, wie man dies programmiert. Man kreiert geeignete Datenstrukturen und definiert konkrete Datentypen. Zum Schluß realisiert man die Operationen als Prozeduren.

Gegenüberstellung: DT, DST, ADT

Lassen Sie uns die drei Begriffe Datentyp, Datenstruktur und abstrakter Datentyp gegenüberstellen: Ein Datentyp legt den Wertebereich fest, den ein Objekt von diesem Typ hat.

Eine Datenstruktur ist die Verbindung mehrerer Werte (Komponenten) zu einer Einheit. Haben die Komponenten den gleichen Datentyp und gleiche Bedeutung, spricht man von einem Feld. Ein Record vereinigt meist verschiedene Datentypen. In einem Feld von Records ist jedes Feldelement ein Record. In Modula 2 kann man eine Datenstruktur als Datentyp definieren und als solchen benutzen. Dies ist in Omikron.BASIC nicht möglich. Die Begriffe Datenstruktur und Datentyp verwendet man oft synonym.

Ein abstrakter Datentyp ist ein mathematisches Modell. Es besteht aus einer abstrakten Typdefinition und Operationen auf diesen Typ. Er ist unabhängig von der Implementierung auf einem Computer. Man realisiert einen abstrakten Datentyp durch geeignete Datentyp- und Datenstrukturen-Definition auf dem zugrundeliegenden Rechner. Auf dieser Basis programmiert man die Operationen als Prozeduren.

Beispiel: ADT 'lineare Liste'

Den Schwerpunkt dieses Kurses bildet der abstrakte Datentyp *lineare Liste*. Eine lineare Liste ist eine verkettete Folge von null oder mehr Elementen eines bestimmten Datentyps. Die Anzahl der Elemente heißt Länge der Liste. Falls sie kein Element enthält, spricht man von einer leeren Liste.

Jede lineare Liste besitzt genau ein erstes und ein letztes Element. Bis auf das letzte hat jedes Element einen Nachfolger. Alle Elemente haben einen Vorgänger, ausgenommen das erste.

Das erste Element nennt man auch Listenkopf.

Sind die Elemente sortiert, spricht man von einer *geordneten linearen Liste*.

Eine lineare Liste ist vergleichbar mit einer (geöffneten) Perlenkette. Jede Perle repräsentiert ein Listenelement, und der Zwirn besorgt die Linearität.

Hat ein Element mehr als einen Vorgänger oder mehr als einen Nachfolger, so ist die Linearität verletzt. Ein solches Gebilde heißt Baum oder Graph.

Betrachten wir beispielsweise die fünf Zahlen 3, 8, 4, 1 und 7. Eine Möglichkeit, diese willkürlich in einer linearen Liste von ganzen Zahlen abzuspeichern, ist 8 -> 4 -> 3 -> 7 -> 1. Das erste Element ist die '8'. Das letzte ist die '1'. Die Länge der Liste ist fünf. '4' ist der Nachfolger von '8'. '3' ist der Vorgänger von '7'.

Eine geordnete lineare Liste mit diesen Zahlen wäre 1 -> 3 -> 4 -> 7 -> 8. In dieser Liste sind '1' und '8' das erste beziehungsweise letzte Element. Die zugrundeliegende Ordnung ist die Kleiner-Relation '<'.

Eine lineare Liste ist ein abstrakter Datentyp. Sinnvolle Operationen sind beispielsweise Suchen, Einfügen, Löschen oder Ausgeben der Liste.

Konzept zur Realisierung des ADT lineare Liste in Modula 2

Ein Listenelement besteht aus zwei Teilen: den Infokomponenten und den Zeigern auf die Nachbarelemente. Die Infokomponenten speichern die Daten. Die Zeiger auf die Nachbarelemente sind Verwaltungsinformationen. Sie besorgen die Verkettung der Liste.

Den abstrakten Datentyp *lineare Liste* realisieren wir in Modula 2 mit zwei konkreten Datentypen:

```
TYPE PersonPtr = POINTER TO
Personenliste;
     Personenliste = RECORD

Name    : String;

Vorname : String;

Alter   : INTEGER;

Telnr   : LONGINT;

Prev    : PersonPtr;

Next    : PersonPtr
END;
```

Personenliste ist eine als Datentyp definierte Datenstruktur. Sie speichert die Daten eines Listenelementes. Die Komponenten *Name*, *Vorname*, *Alter* und *Telnr* sind die Infokomponenten. Hier speichern wir die Daten der einzelnen Personen. *Prev* und *Next* sind Verwaltungsinformationen - sie besorgen die Listeneigenschaft.

Der Datentyp *PersonPtr* definiert einen Pointer (Zeiger) auf ein Listenelement. Ein Wert vom Typ *PersonPtr* ist ein Zeiger auf ein Objekt vom Typ *Personenliste*. *Prev* (Previous) bezeichnet den Zeiger auf das Vorgängerelement, *Next* den Zeiger auf den Nachfolger.

Bitte beachten Sie die Verstrickung der beiden Datentypen. *PersonPtr* ist mit Hilfe von *Personenliste* definiert und umgekehrt.

Einen Datentyp zur Speicherung von ganzen Zahlen in einer linearen Liste definieren Sie folgendermaßen:

```
TYPE ZahlenPtr = POINTER TO
                 Zahlenliste;
     Zahlenliste = RECORD
     Zahl   : INTEGER;
     Next   : ZahlenPtr
END;
```

Die Listenelemente sind vom Datentyp *Zahlenliste*. *ZahlenPtr* definiert einen Zeiger auf ein Objekt vom Datentyp *Zahlenliste*.

Entsprechend definieren wir einen Datentyp zur Speicherung von Wörtern in einer linearen Liste. Diese Liste nennen wir *Wortliste*.

```
TYPE WortPtr = POINTER TO Wort-
                          liste;
     Wortliste = RECORD
     Wort   : String;
     Next   : WortPtr
END;
```

Rekursiver Datentyp

Beide Listenstrukturen bezeichnet man als *rekursive Datentypen*, da sie Zeiger auf sich selbst enthalten. Rekursive Datentypen lassen sich einfach mit rekursiven Algorithmen bearbeiten. Bei linearen Listen ist Rekursion meist überflüssig. Für Baumstrukturen bietet sie sich an.

Im Datentyp *Personenliste* haben wir zwei Zeiger definiert: einen auf den Vorgänger und einen auf den Nachfolger. Die so entstehende Listenstruktur bezeichnet man als *doppelt verkettete* lineare Liste. Enthält eine Listenstruktur nur einen Zeiger auf das Nachfolgeelement, spricht man von einer *einfach verketteten* linearen Liste. *Zahlenliste* ist ein Beispiel für diesen Typ.

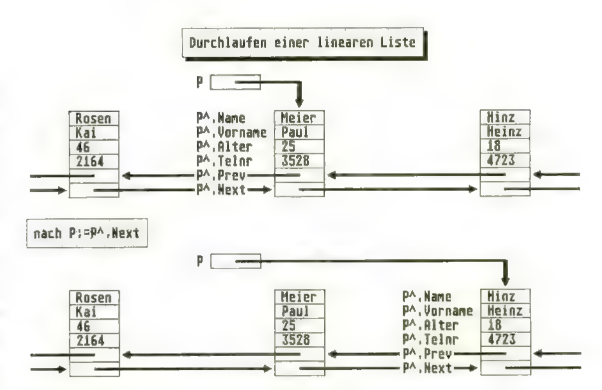
Bild 1 stellt beide Listentypen grafisch dar. Die Infokomponenten sind grau unterlegt. Da sie für die Listenstruktur ohne Bedeutung sind, packen wir alle Infokomponenten zusammen in eine Box. Zeiger (Pointer) zeichnet man allgemein als Pfeile. Sie zeigen auf die gesamte Datenstruktur und nicht auf eine einzelne Komponente. Die Pfeilrichtung ist wesentlich - Zeiger sind „Einbahnstraßen".

Das durchgestrichene Zeigerfeld symbolisiert den Listenanfang bzw. das Listenende. Dieser Zeiger hat einen besonderen Namen: *NIL*. Ein *NIL*-Zeiger zeigt auf *nichts* - sein Wert ist undefiniert.
Bild 1 enthält zwei weitere Zeiger: *Anfang* und *Ende*. Der Zeiger *Anfang* identifiziert den Anfang der Liste. Entsprechend verweist *Ende* auf das Listenende.

Anfang ist in einfach verketteten Listen unersetzlich. *Ende* ist Luxus - das letzte Element finden Sie auch mittels Durchlaufen der Liste.

Bei doppelter Verkettung benötigen Sie mindestens einen Zeiger auf ein beliebiges Listenelement. Im Gegensatz zu einfach verketteten Listen können Sie eine Liste auch rückwärts durchlaufen. Von jedem Element gibt es einen Pfad zu jedem anderen Element, also auch zum Listenanfang und -ende. Sinnvollerweise wählt man einen Zeiger auf den Listenkopf oder das Ende zur Identifikation.

Allgemein charakterisiert man den Anfang einer linearen Liste durch einen Zeiger auf das erste Element und nicht durch das erste Listenelement selbst. *Anfang* ist beispielsweise vom Datentyp *PersonPtr*, nicht vom Typ *Personenliste*.



Dynamische Speicherverwaltung, Allocate

Ein Vorteil von Listen ist der dynamische Speicherplatzbedarf. Der Speicherplatz für ein Listenelement wird erst angefordert, wenn die Daten gespeichert werden; beispielsweise wenn der Benutzer Daten über die Tastatur eingibt. Dadurch wächst und schrumpft der Speicherbedarf einer Liste zur Programmlaufzeit - wie gesagt: dynamisch.

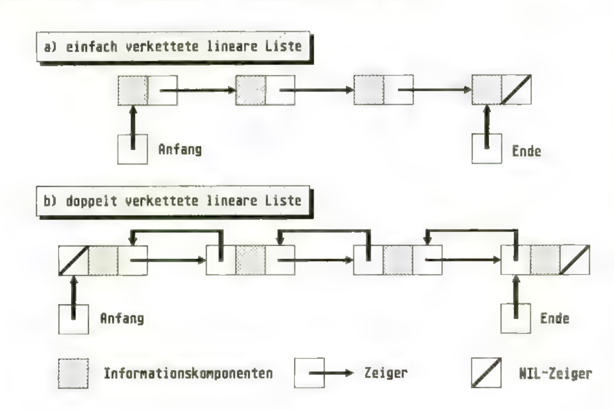
Den Speicherplatz für ein Objekt eines bestimmten Datentyps besorgt zur Programmlaufzeit die Prozedur *Allocate(P,TSIZE(<Datentyp>))*. *Allocate* reserviert im Speicher soviel Speicherplatz, wie wir zur Speicherung eines Wertes des gewünschten Datentyps benötigen. In der Zeigervariablen *P* gibt sie einen Zeiger auf diesen Wert zurück.

Umgekehrt gibt *Deallocate(P,TSIZE(<Datentyp>))* den Speicherplatz des Objekts frei, auf das *P* zeigt.

Zugriff auf die Listenelemente, Durchlaufen einer Liste

Die Variable *P* sei ein Zeiger auf einen Wert eines bestimmten Datentyps, beispielsweise vom Typ *PersonPtr*. Mit dem Dereferenzieroperator '^' greifen wir auf die Datenstruktur zu, auf die *P* zeigt. *P^.Name* ist der Name der Person. Mit *P^.Alter* adressieren Sie die Alterskomponente. *P^.Next* ist wiederum ein Zeiger - auf den Nachfolger. Die Zuweisung *P:=P^.Next* läßt den Zeiger *P* auf den Nachfolger zeigen. Auf diese Weise durchläuft *P* eine lineare Liste. Bild 2 demonstriert dies. Den Operator '^' wenden Sie nur an, wenn *P* ein Zeiger ist.

Mit unserem jetzigen Wissen sind wir in der Lage, eine lineare Liste in Modula 2 zu bearbeiten. Der Zeiger *Anfang* auf das erste Listenelement identifiziert die Liste. *Ende* zeigt auf das Listenende. *Anfang* und *Ende* seien vom Datentyp *ZahlenPtr*. Die folgenden Anweisungen bilden eine lineare Liste aus den vier Zahlen 4, 6, 1 und 9. Die Zahlen werden während des Aufbaus an die bis dahin bestehende Liste angehängt:




```
Anfang:=NIL; Ende:=NIL; (* leere
                           Liste am Anfang *)
(* '4' wird erstes Listenelement *)
Allocate(Anfang,TSIZE(Zahlenliste));
Anfang^.Zahl:=4;
Ende:=Anfang;
Allocate(Ende^.Next,TSIZE(Zahlenliste));
                           (* '6' anhängen *)
Ende:=Ende^.Next;
Ende^.Zahl:=6;
Allocate(Ende^.Next,TSIZE(Zahlenliste));
                           (* '1' anhängen *)
Ende:=Ende^.Next;
Ende^.Zahl:=1;
Allocate(Ende^.Next,TSIZE(Zahlenliste));
                           (* '9' anhängen *)
Ende:=Ende^.Next;
Ende^.Zahl:=9;
Ende^.Next:=NIL; (* das Listenende
                      markieren *)
```

In der Praxis konstruiert man Listen nicht derart schrittweise. Wir haben diese Form gewählt, um Sie nicht mit dem Modula 2-spezifischem Umfeld zu belasten.

Bei der Übertragung des Listenkonzepts auf Omikron.BASIC bereitet schon die Typdefinition Probleme: Omikron.BASIC kennt keinen Datentyp Zeiger. Wie realisieren wir also die Verkettung von Listenelementen? Modula 2-Zeiger sind Speicheradressen zur Programmlaufzeit. Das heißt, die *Next*-Komponente eines Elementes speichert die Adresse des nächsten Listenelementes. Einen solchen Zeiger bezeichnen wir als *Pointer*. Das Pointer-Konzept läßt sich nur mit großem 'Fummel'-Aufwand nach Omikron.BASIC portieren.

Eine einfache Alternative zu Adressen sind Feldindizes: Wir speichern die lineare Liste in Feldern. Für die *Next*-Komponente dimensionieren wir ein Feld ganzer Zahlen zur Speicherung der Zeiger. Eine solche Zahl interpretieren wir als Feldindex des nächsten Elements. Einen Index, der als Zeiger verwendet wird, nennt man in der Informatik einen *Cursor*.

Unsere Personendaten speichern wir als Liste in Omikron.BASIC beispielsweise in den Feldern:

```
Personen_Name$()
Personen_Vorname$()
Personen_Alter%()
Personen_Telnr%L()
Personen_Prev%()
Personen_Next%()
```

Ein Element der Felder *Personen_Prev%()* und *Personen_Next%()* speichert jeweils einen Cursor.

Bild 3 zeigt eine alphabetisch geordnete Personenliste sowohl als Cursor-Implementierung als auch unter Verwendung von Pointern.

Das *NIL*-Element markiert in Modula 2 das Listenende. Bei Cursor-Verwendung realisieren wir *NIL* durch den Index 0. Daraus folgt, daß die Feldelemente mit den Indizes Null keine Daten speichern. Ihr Wert ist undefiniert. Dies ist wesentlich, da wir später die Null-Elemente zur Optimierung einiger Algorithmen benutzen.

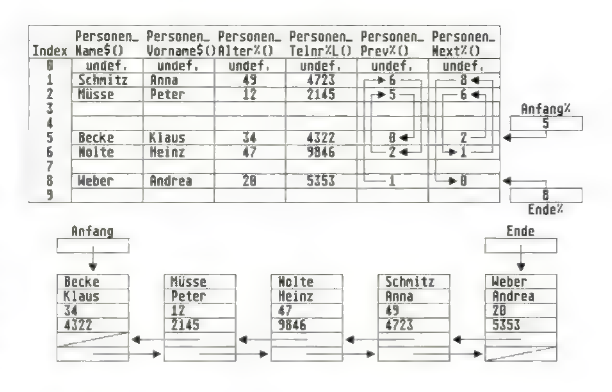
Es ist unwichtig, wo die Daten im Feld gespeichert sind. Allein durch die *Prev*- und *Next*-Zeiger rekonstruieren wir die Reihenfolge. Bei der grafischen Darstellung eines Cursors schreibt man den Zahlenwert des Zeigers zusätzlich zum Pfeil in das Zeigerfeld. Bei Pointern zeichnet man nur den Pfeil.

Die Felder können mehrere lineare Listen speichern. Jede Liste benötigt einen Cursor auf den Listenanfang. Bild 4 demonstriert dies an den drei Wortlisten

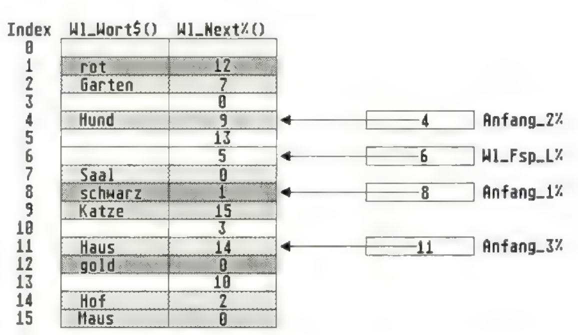
```
'schwarz' -> 'rot'   -> 'gold'
'Hund'    -> 'Katze' -> 'Maus'
'Haus'    -> 'Hof'   -> 'Garten'
                     -> 'Saal'
```

Diese speichern wir in den Feldern *Wl_Wort$()* und *Wl_Next%()*, gemäß der oben beschriebenen Datenstruktur *Wortliste*. Die verschiedenen Grautöne verdeutlichen, welche Elemente in welche Liste gehören. Die Pfeile für die Cursor innerhalb des Feldes haben wir aufgrund der Übersichtlichkeit weggelassen.

Wl_Fsp_L% ist ein Cursor auf die sogenannte *Freispeicherliste* für das Record-Feld *Wl*. Die Freispeicherliste listet alle freien Feldelemente auf. Die Werte in den freien Elementen sind undefiniert. Dort steht ein beliebiger Wert.



Index	Personen_ Name$()	Personen_ Vorname$()	Personen_ Alter%()	Personen_ Telnr%L()	Personen_ Prev%()	Personen_ Next%()
0	undef.	undef.	undef.	undef.	undef.	undef.
1	Schmitz	Anna	49	4723	6	8
2	Müsse	Peter	12	2145	5	6
3						
4						
5	Becke	Klaus	34	4322	0	2
6	Nolte	Heinz	47	9846	2	1
7						
8	Weber	Andrea	20	5353	1	0
9						



Index	Wl_Wort$()	Wl_Next%()
0		
1	rot	12
2	Garten	7
3		0
4	Hund	9
5		13
6		5
7	Saal	0
8	schwarz	1
9	Katze	15
10		3
11	Haus	14
12	gold	0
13		10
14	Hof	2
15	Maus	0

Freispeicherverwaltung: Allocate, Deallocate

Wie in Modula 2 benötigen wir Prozeduren zur Bereitstellung und Freigabe von Speicherplatz für Listenelemente. Wir schreiben uns dazu zwei Prozeduren: *Allocate_xxx* und *Deallocate_xxx*. Beide sind abhängig von der Datenstruktur, die den Listenelementen zugrundeliegt. *xxx* steht im Prozedurnamen für den Namen der Datenstruktur. In unseren Beispielen nennen wir die Prozeduren *Allocate_Personen()* und *Deallocate_Personen()* beziehungsweise *Allocate_Wl()* und *Deallocate_Wl()*. Mit *Wl* kürzen wir *Wortliste* ab, da Prozedur- und Variablennamen sonst zu lang sind.

Allocate_xxx(R P%) liefert einen Cursor *P%* auf ein freies Element. Falls kein Eintrag frei ist, wird *P%* gleich Null (NIL).

Deallocate_xxx(P%) markiert das Element als unbelegt, auf das der Cursor *P%* zeigt.

Sven Krüppel

Literatur:

Niklaus Wirth, Algorithmen und Datenstrukturen mit Modula 2, B.G. Teubner Stuttgart

Aho, Hopcroft, Ullman, Data Structures and Algorithms, Addison Wesley

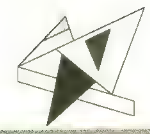
Kurt Mehlhorn, Datenstrukturen und effiziente Algorithmen, Band 1 Sortieren und Suchen, B.G. Teubner Stuttgart

```
 1:'************************************************
 2:'*                                              *
 3:'* Programm zur Verwaltung des abstrakten       *
              Datentyps 'Person'                   *
 4:'*                                              *
 5:'* (c) MAXON Computer 1990    by Sven Krüppel   *
 6:'*                                              *
 7:'************************************************
 8:
 9:'  Initialisierungen d.Variable Personendaten
    v.abstrakten Datentyp Person
10:   Personendaten_Maxanzahl#=100' Maximale
    Personenzahl, die gespeichert wird
11:   DIM Personendaten_Name$(Personendaten_
                              Maxanzahl#-1)
12:   DIM Personendaten_Vorname$(Personendaten_
                                 Maxanzahl#-1)
13:   DIM Personendaten_Alter%(Personendaten_
                               Maxanzahl#-1)
14:   DIM Personendaten_Telnr%L(Personendaten_
                                Maxanzahl#-1)
15:
16:   'Hauptprogramm (nur zu Test- und
               Demonstrationszwecken)
17:   Speichere_Personendaten(0,"Mustermann","Paul",
                              45,4156)
18:   Speichere_Personendaten(1,"Schuster","Klara",
                              38,6138)
19:   Speichere_Personendaten(2,"Hermanns","Ira",18,
                              2345)
20:   Kopiere_Personendaten(2,3)
21:   Speichere_Personendaten(3,"","Tanja",20,0)' Die
    ältere Schwester von Ira
22:   Vertausche_Personendaten(0,1)
23:
24:   FOR I%=0 TO 3
25:     Lies_Personendaten(I%,N$,V$,A%,T%L)
26:     PRINT N$,V$,A%,T%L
27:   NEXT I%
28:   END
29:
30:'*********************************************
31:' Der Zugriff auf die Personendaten sollte nur
32:'      über die folg. Prozeduren erfolgen.
33:'*********************************************
34:
35:   DEF PROC Speichere_Personendaten(Index%,Name$,
                  Vorname$,Alter%,Telnr%L)
36:   ' Schreibt die übergebenen Werte in die
      Elemente mit dem Index Index% von Feld
37:   ' Personendaten. Leer-Strings bei den String-
      Komponenten und der Wert Null
38:   ' b.d.Zahlenwerten bedeuten, daß d.Komponente
      nicht verändert wird.
39:
40:     IF Index%>=0 AND Index%<Personendaten_
      Maxanzahl# THEN
41:        IF Name$<>"" THEN Personendaten_Name$
        (Index%)=Name$
42:        IF Vorname$<>"" THEN Personendaten_
        Vorname$(Index%)=Vorname$
43:        IF Alter%<>0 THEN Personendaten_Alter%
        (Index%)=Alter%
44:        IF Telnr%L<>0 THEN Personendaten_Telnr%L
        (Index%)=Telnr%L
45:     ELSE
46:        PRINT "Index außerhalb des zulässigen
                 Bereiches"
47:     ENDIF
48:   RETURN ' Speichere_Personendaten
49:
50:
51:   DEF PROC Lies_Personendaten(Index%,R Name$,R
                  Vorname$,R Alter%,R Telnr%L)
52:   ' Gibt die Werte in den Komponenten des
      Elementes mit dem Index Index%
53:   ' von Feld Personenendaten zurück.
54:
55:     IF Index%>=0 AND Index%<Personendaten_
      Maxanzahl# THEN
```

```
56:        Name$=Personendaten_Name$(Index%)
57:        Vorname$=Personendaten_Vorname$(Index%)
58:        Alter%=Personendaten_Alter%(Index%)
59:        Telnr%L=Personendaten_Telnr%L(Index%)
60:     ELSE
61:        PRINT "Index außerhalb des zulässigen
                 Bereiches"
62:     ENDIF
63:   RETURN ' Lies_Personendaten
64:
65:
66:   DEF PROC Kopiere_Personendaten
    (Quell_Index%,Ziel_Index%)
67:   ' Kopiert sämtliche Komponenten an Stelle
      Quell_Index% nach Ziel_Index%
68:
69:     IF Quell_Index%>=0 AND Quell_Index%<
      Personendaten_Maxanzahl#
        THEN
70:       IF Ziel_Index%>=0 AND Ziel_Index%<
        Personendaten_Maxanzahl# THEN
71:          Personendaten_Name$(Ziel_Index%)=
                Personendaten_Name$(Quell_Index%)
72:          Personendaten_Vorname$(Ziel_Index%)=
                Personendaten_Vorname$(Quell_Index%)
73:          Personendaten_Alter%(Ziel_Index%)=
                Personendaten_Alter%(Quell_Index%)
74:          Personendaten_Telnr%L(Ziel_Index%)=
                Personendaten_Telnr%L(Quell_Index%)
75:       ELSE
76:         PRINT "Zielindex außerhalb des zulässigen
                   Bereiches"
77:       ENDIF
78:     ELSE
79:       PRINT "Quellindex außerhalb des zulässigen
                 Bereiches"
80:     ENDIF
81:   RETURN ' Kopiere_Personendaten
82:
83:
84:   DEF PROC Vertausche_Personendaten(Index1%,
                                        Index2%)
85:   ' Vertauscht sämtliche Komponenten der Stellen
      Index1% und Index2%
86:
87:     LOCAL Name$,Vorname$,Alter%,Telnr%L
88:     IF Index1%>=0 AND
           Index1%<Personendaten_Maxanzahl# THEN
89:       IF Index2%>=0 AND
           Index2%<Personendaten_Maxanzahl#
          THEN
90:         Name$=Personendaten_Name$(Index2%)
91:         Vorname$=Personendaten_Vorname$(Index2%)
92:         Alter%=Personendaten_Alter%(Index2%)
93:         Telnr%L=Personendaten_Telnr%L(Index2%)
94:
95:         Personendaten_Name$(Index2%)=
                 Personendaten_Name$(Index1%)
96:         Personendaten_Vorname$(Index2%)=
                 Personendaten_Vorname$(Index1%)
97:         Personendaten_Alter%(Index2%)=
                 Personendaten_Alter%(Index1%)
98:         Personendaten_Telnr%L(Index2%)=
                 Personendaten_Telnr%L(Index1%)
99:
100:        Personendaten_Name$(Index1%)=Name$
101:        Personendaten_Vorname$(Index1%)=Vorname$
102:        Personendaten_Alter%(Index1%)=Alter%
103:        Personendaten_Telnr%L(Index1%)=Telnr%L
104:      ELSE
105:        PRINT "Index2 außerhalb des zulässigen
                 Bereiches"
106:      ENDIF
107:    ELSE
108:      PRINT "Index1 außerhalb des zulässigen
               Bereiches"
109:    ENDIF
110:  RETURN ' Vertausche_Personendaten
```



JETZT UMSTELLEN



Die professionelle Abrechnungs-Software für Ihren ST

CIS-Lohn & Gehalt ist das derzeit einzige durch die Spitzenverbände der Sozialversicherer zum automatisierten DEVO-Meldeverfahren zugelassene Programm für Atari-ST-Computer. *Auszug aus dem Sachregister:* Schnittstellen zu TIM II, ST-Fibu, fibuMan, Berlin-Abrechnung, Baulohn, Kassenlisten, Meldewesen, Steuerberechnung, modulares Konzept, Urlaubskonto, Sonderauswertungen, betr. Altersversorgung, VL, ZVK, Urlaubskasse ...



CIS - Es ist Ihre Entscheidung, wieviel Arbeit Sie sich durch unsere Produkte sparen.

Ciechowski Computer Innovations Ober-Saulheimer Straße 18, D-6501 Wörrstadt, Telefon (06732) 7354 oder 5018

STÜCKLISTEN FAKTURIERUNG * HOBBY BUCHFÜHRUNG * LERNEN

verwalten mit K-FAKT ST (Fakturierung)

COMPTABLE ST (Buchführung) DM 198.00
Ein Buchführungsprogramm für Geschäft und Privat. 500 Konten (DATEV), 10 Steuersätze, bis zu 10000 Buchungen. Steuersätze und Privatanteilsätze können bereits im Kontenrahmen vorgegeben werden. Auswertungen: Saldenliste, Kontenblätter, Journal, Kassenbuch, Gewinn/Verlust-Rechnung, Umsatzsteuerdaten jeweils für einen beliebigen Monat, ein beliebiges Quartal oder das ganze Jahr! Alle Ausgaben auf Bildschirm, Drucker oder Datei. Buchungsmemo für 25 Buchungen (einfach durch Anklicken abrufbar). Universelle Druckeranpassung. Incl. 50-seitigem, reichhaltig illustriertem Handbuch (mit Ausdruckbeispielen) im Ringordner! DEMO-DISKETTE DM 20.--

K-FIBU ST (Kleine Finanzbuchhaltung) DM 398.00
Leistungen wie COMPTABLE ST, zusätzlich Bilanz, erweiterte Umsatzsteuerdaten-Auswertung und Modulschnittstelle für Buchungsimport (z.B. von K-FAKT ST). DEMO-DISKETTE DM 20,--, HANDBUCH VORAB DM 50,- (Handbuch wird angerechnet)

K-FAKT ST (Professionelles Fakturierungssystem) DM 398.00
Adress-, Artikel- und Stücklistenverwaltung, Umsatz/Absatzstatistik, frei wählbare Indexfelder. Erstellung von Angebot, Auftr.Best., Lieferschein, Rechnung, Storno, Gutschrift, Mahnung, Anfrage, Bestellung, Eingangsbeleg, Aufmaßberechnungen. Alle Arten von Listen und Formularen. Anbindung an 1st Word Plus. Besteuerung nach Artikel oder Kunde, somit auch für Steuersystem in der Schweiz geeignet. Universeller Druckertreiber, an alle Drucker anzupassen, incl. ausführlichem, über 100 Seiten starken Handbuch mit zahlreichen Illustrationen. DEMO-VERSION DM 20. HANDBUCH VORAB DM 50 (wird angerechnet)

ST-MATHETRAINER II (Lernprogramm) DM 59.00
Für 1.-6. Schuljahr. 1x1 und Mischrechnen mit wählbaren Höchstzahlenwerten, Umrechnung von Gewichten und Längenmaßen, im Schwierigkeitsgrad durch eingebaute Editierfunktion frei an den Lernbedarf bzw. den Schulbuchstoff anzupassen. Mit Benotung und Protokoll. Voll in GEM eingebunden! Incl. deutscher Bedienungsanleitung.

ST-RECHTSCHREIBEN II (Lernprogramm) DM 59.00
Für 1.-6. Schuljahr. Wörter in Sätze einfügen, Singular und Plural, Kommata setzen. Im Schwierigkeitsgrad durch eingebaute Editierfunktion frei an den Lernbedarf bzw. Schulbuchstoff anzupassen. Mit Benotung und Protokoll. Voll in GEM eingebunden! Incl. deutscher Bedienungsanleitung.

TKC-VIDEO (Verwaltung von Videocassetten) DM 79.00
Verwaltet bis zu 5000 Videofilme pro Datei. 10 Felder für Titel, Spielzeit, Zählwerk, Darsteller, 'verliehen an', Bemerkungen usw. Umfangreiche Such- u. Selektierfunktionen. Ausdruck von Listen + Etiketten (frei über ASCII-Maske zu gestalten). Voll unter GEM, deutsche Bedienungsanleitung

TKC-MUSICBOX (Musiktitelverwaltung) DM 79.00
Verwaltet bis zu 5000 CDs, LPs oder MCs. Suche nach Einzeltiteln, Ausdruck von Haupttitelübersicht u. Gesamttitelliste mit MC-Nummern und Etiketten. Voll unter GEM, incl. deutscher Bedienungsanleitung

TKC-TRAINER (Trainingsprogramm) DM 99.00
Sehr hoher Lernerfolg durch Karteikastenprinzip. Für Sprachen, Chemie, Physik etc. Voll unter GEM, leicht auch für Schüler zu bedienen. Incl. deutschem Handbuch

TKC-BANKMANAGER (mit Sammler-Ausdruck) DM 99.00

ST-GIRO PLUS DM 49.00
Überweisungsdruckprogramme für den geschäftlichen (häufigen) und privaten Gebrauch. Auch für Schecks + Lastschriften. Ausdruck über ASCII-Masken an alle Formulare anzupassen.

ST-VOKABELTRAINER (Trainingsprogramm) DM 49.00
Vokabeltraining für Englisch, Französisch, Italienisch, Spanisch und zeichenkompatible Sprachen. 100 Vokabeln pro Datei, voll unter GEM

TK COMPUTER-TECHNIK Thomas Kaschadt
Hard- & Software - Entwicklung & Vertrieb

Bischofsheimer Straße 17, D-6097 Trebur-Astheim, Fernruf: (06147) 3550 * Postfach 60, FAX 06147-3555, Btx. 06147-3555

24-Stunden-Auftragsannahme per Anrufbeantworter!

WONDERLAND

Wonderland
Computerservice Höger
Postfach 1051
6912 Dielheim bei Walldorf
Tel. 06227-63587

☆☆☆☆☆☆☆☆☆☆☆

Unser Weihnachtsangebot für alle, die ihren ATARI ST Computer AT-kompatibel machen wollen und dazu Emulator plus Festplatte benötigen:

DataJet-AT
40 MB Festplatten-Subsystem DataJetX40
plus AT-Emulator ATonce — Nr. 0704 DM 1.398,-

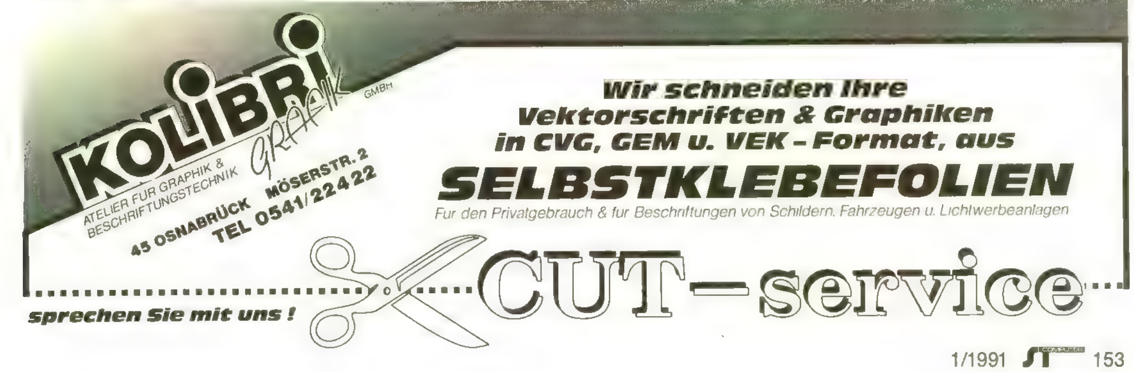
☆☆☆☆☆☆☆☆☆☆☆

vortex ATonce

AT-Emulator für Atari ST und Mega ST	Nr. 0900	DM	428,-
Steckadapter für Mega ST	Nr. 0901	DM	69,-
Steckadapter für 1040 STE	Nr. 0902	DM	89,-

Wonderland bietet ausschließlich original vortex Produkte.
Die Festplatten sind anschlußfertig und betriebsbereit mit sämtlichen Anschluß-Kabeln, System-Software und deutschem Benutzer-Handbuch bzw. Betriebsanleitung.
Volle Gewährleistung. Zehn Tage uneingeschränktes Rückgaberecht.
24 Stunden/7 Tage Bestellservice: Anrufbeantworter oder Postkarte/Brief
Mail-Order-Versand: Keine Beratung, kein Ladenverkauf.
Lieferung nach Vorauskasse (Euroscheck max. DM 400,-) oder Post-Nachnahme: Sofort.
Eingang der Ware bei Ihnen innerhalb einer Woche.
Versandkosten pro Lieferung innerhalb BRD DM 10,--.
Auslandsbestellungen nur gegen Vorauskasse: Kosten nach Aufwand.

NEU NEU NEU NEU NEU NEU NEU NEU NEU NEU NEU NEU NEU NEU NEU



nützliche SOFTWARE für alle Naturwissenschaftler



Logiksimulator für den ATARI ST
Ein Programm zum Erstellen, Testen und Analysieren von Logikschaltungen für Ausbildung und Hobbyelektronik
- komfortable GEM-Umgebung/Bauteile lassen sich per Maus plazieren und verdrahten/Umfassendes Bauteile-Setz (Grundgatter, Ein/Ausgabe-Bausteine,Flip-Flops)/ Definition zusätzlicher Bauteile durchMakrotechnik/ Makros können in Libraries gespeichert werden/Erzeugen von Impulsdiagrammen/Hardcopy-Funktion/Druckertreiber

NEU ab VERSION 2.0
- vergrößerter Arbeitsspeicher (4 Bildschirme)/Gruppenfunktion Editor (verschieben, duplizieren, löschen)/Ausgabe von Schaltbildern und Diagrammen in Bilddatei/erweiterte Druckfunktion (ganzes Bild im Querformat)/zusätzliche Druckertreiber vorhanden/Simulation (Berücksichtigung von Bauteilschaltzeiten, Einzelschrittausführung)/16-BIT-Interface - Signal Ein/Ausgabe - Logikanalyse - Bauanleitung steht im Handbuch

DM 89,--

Simulation von ANALOGSCHALTUNGEN
Das Simulationsprogramm zum Analysieren, Testen und Entwickeln von analogen Elektonikschaltungen (Kettenschaltungen) für Hobby, Ausbildung und Studium.
- komfortable Maussteuerung/Grafischer Schaltungsaufbau/ einfache Eingabe und Angabe von Bauteilwerten/max 65 Bauteile pro Schaltung (Stromquellen, Übertrager, Schwingkreise, offene und kurzgeschlossene Übertragungsleitungen usw.)/Wobbel-Generator von 1HZ bis 2 GHZ/max 0,1 HZ bzw. 1 HZ Auflösung/Wahlmöglichkeit zwischen sehr schneller oder sehr genauer Berechnung/grafische Ausgabe von Spannungs- und Stromverhältnissen, von Phasenverläufen und Eingangsimpedanzen/logarithmische und lineare Koordinatenachsen/Verstärkung und Offset einstellbar/ einfaches Testen der Schaltung im Rückwärtsbetrieb / Hardcopy-Funktion

DM 98,--

ST Analog



Simulation elektrischer Felder auf dem ST
Diese Software ersetzt den aus der Physik bekannten Versuch mit dem Eisenfeilspahn. Besonders interesannt ist das Programm für Schulen und Studenten der Physik oder Elektrotechnik. Probleme aus der Literartur können mit dem Computer nachempfunden werden. Die Konstelation der Ladungen, sowie deren Stärke ist schnell zu ändern und somit ohne Probleme auf dem Monitor sichtbart zu machen. Die Eigenschaften von ST-Feld:
- benutzerfreundliche, mausgesteuerte Arbeitsoberfläche/Darstellung von Feldlinien und Äquipotentialinien/bis zu 20 Ladungen können auf dem Bildschirm plaziert werden/setzen mit der Maus oder Direkteingabe der Position möglich/Weiterverarbeitung mit Text oder Grafikprogrammen möglich

benötigt s/w-Monitor

DM 98,--

Projektplanungssystem für Netzpläne, Balkendiagramme und Terminlisten. Es übernimmt die bei Netzplänen zeitaufwendige Rechen- und Zeichenarbeit. Das Programm ist für profesionelle und semiprofessionelle Anwender geeignet. Es ermöglicht eine detaillierte Analyse aller Projekte und somit eine Optimierung ihrer Projekte, denn Zeit ist Geld.
Highlights von ST-Netzplan:
Termin- und Ablaufplanung/GEM-gesteuert/500 Vorgänge je Netzplan/Projektlänge bis 999 Arbeitstage/Kalender bis Jahr 2099/Plausibilitätsprüfung bei der Eingabe/einstellbare Feiertage/kompatible Druckeransteuerung für EPSON und NEC kompatible Drucker/ON-Line Help Funktion

DM 98,--





Zeichenprogramm für chemische Moleküle
läuft auf allen ATARI ST mit 1MB und s/w-Monitor
- voll GEM unterstützt/48 Pictogramme erlauben einen schnellen Aufruf fast aller Funktionen/beliebiges Papierformat einstellbar/Druckertreiber für 9 und 24 Nadeldrucker sind vorhanden/Grafikausgabe ist in drei Pixelformaten möglich (Degas, Doodle, STAD)/Moleküle können gedreht, gestaucht, gespiegelt, vergrößert und verkleinert werden/ Viele vorgefertigte Moleküle vorhanden/6 Einfachbindungsarten/3 Doppelbindungsarten/14 Pfeilarten/Griechicher und lateinischer Plottzeichensatz/Kopier- und Verschiebefunktion/ Schneller Bildschirmaufbau/Polynomschlange/Quickmaus/ festes Fadenkreuz/CPIXEL-Dienstprogramm zum direkten Einbinden der Grafiken in eine Textverarbeitung/

V.1.1F läuft auch unter TOS 1.4

DM 98,--

System- und Regelungsanalyse auf dem ST
MEKMALE:
- uneingeschränkte Nullstellensuche (komplex und reel)/ Verwaltung von drei Funktionen gleichzeitig/arbeitet in der Bode-Normalform und in der Polynomdarstellung/beherscht Partialbruchzerlegung und Rücktransformation/beherrscht alle gängigen Regelkreisglieder/-strecken/ eigene Regelkreis glieder/-strecken definierbar/grafische Ausgabe der Zeitantwort bei verschiedenen Anregungen/Kreisschließung/jede Darstellung mit vielen Extafunktionen/grafische Ergebnisse der Auswertungen können in anderen Programmen weiterverwendet werden/voll GEM-unterstützt/frei wählbare Ausgabemöglichkeit (Monitor, Drucker, Disk)/erstellte Grafiken können nachträglich weiterbearbeitet werden/beherrscht alle gängigen Darstellungsformen wie: Wurzelortdarstellung, Nyquist-Kurve, Nichols-Diagramm, Nichols-Pläne, 3D-Relief, Sigma-Bode-Diagramm, jw-Bode-Diagramm und Block-Diagramm

Version 2.3 **DM 198,--**



Preise sind unverbindlich empfohlene Verkaufspreise

BESTELL - COUPON

Bitte senden Sie mir

____

____

zzgl. Versandk. DM 6,--
(Ausland DM 10,--)
unabhängig von der bestellten Stückzahl

Name, Vorname ______________________

Straße ______________________

PLZ, Ort ______________________
oder benutzen Sie eingeheftete Bestellkarte

Heim Verlag
Heidelberger Landstr.194
6100 Darmstadt-Eberstadt
Telefon: 06151/560957-58
Telefax: 06151/56059

in Österreich:
RRR EDV GmbH
Dr. Stumpf Str. 118
A-6020 Insbruck

in der Schweiz:
Data Trade AG
Landstr.1
CH-5415 Rieden-Baden

The Return Of



The Forgotten Cookies

Natürlich pünktlich zur Weihnachtsausgabe ist es mal wieder passiert. In der Layoutabteilung hat man vor lauter verzweifeltem Überlegen, was die lieben Verwandten und Bekannten bisher unterm Weihnachtsbaum vermißt haben, die beiden Listings unseres CookieJar-Artikels in der letzten Ausgabe (12/90) vergessen. Da diese Ausgabe aber vor Weihnachten auf dem Markt kommt, fällt dieses nachgereichte Gebäck auch noch in die Adventszeit und somit in die tradionelle Hochsaison aller Backspezialisten. Wohl bekomm's und guten Appetit!

```
 1:  /*
 2:      So könnte ein Programm überprüfen, ob es
 3:      bereits installiert ist; auf diese Weise
 4:      lassen sich Mehrfach-Installationen leicht
 5:      verhindern...
 6:
 7:      entwickelt mit Turbo-C (Version 2.0)
 8:  */
 9:
10:
11:  #include <stdio.h>
12:
13:  /*  Hier sollte die Header-Datei mit den
14:      Prototoypen des Cookie-Moduls einge-
15:      lesen werden.
16:  */
17:
18:
19:  void main (void)
20:  {
21:      COOKIE test;
22:      long wert = 0;
23:
24:      if (!get_cookie („XXXX", &wert))
25:      {
26:          /*  wir sind noch nicht da !  */
27:          create_cookie (&test, „XXXX", 0L);
28:          new_cookie (&test);
29:
30:          /*  hier geht's weiter...   */
31:      }
32:
33:      else
34:      {
35:          /*  sind bereits im System  */
36:          puts („Hurra - Cookie gefunden !");
37:          printf („Cookie-Wert: %lx", wert);
38:      }
39:  }
```

Listing 1

```
 1:  /*
 2:     Programm  : Extended TOS Library
 3:     Modul     : cookie.c
 4:     Funktion  : allgemeine Routinen zur Verwaltung
 5:                 von Cookie Jars
 6:
 7:     Edit      : 01.04.90
 8:     Autor     : Rolf Kotzian
 9:
10:     Copyright: (c) MAXON Computer GmbH
11:  */
12:
13:
14:
15:  /*  Includes  */
16:
17:  #include <stdio.h>
18:  #include <stdlib.h>
19:  #include <string.h>
20:  #include <tos.h>
21:
22:
23:  /*  Defines  */
24:
25:  #define TRUE                      (0 == 0)
26:  #define FALSE                     (0 == 1)
27:
28:
29:  /*  Typ-Vereinbarungen  */
30:
31:  typedef int boolean;
32:  typedef struct
33:  {
34:      char cookie_id[4];        /*  Cookie-ID    */
35:      long cookie_value;        /*  Cookie-Wert  */
36:
37:  }COOKIE;
38:
39:
40:
41:  /*
42:      CREATE_COOKIE:  installiert in der Variablen
43:                      'cookie' einen Cookie; als
44:                      weitere Parameter werden die
45:                      ID des Cookies sowie dessen
46:                      Wert übergeben.
47:  */
48:
49:  void create_cookie (COOKIE *cookie, char *id,
50:                      long value)
51:  {
52:      strncpy (cookie->cookie_id, id, 4);
53:      cookie->cookie_value = value;
54:
55:  }  /*  create_cookie  */
56:
57:
```

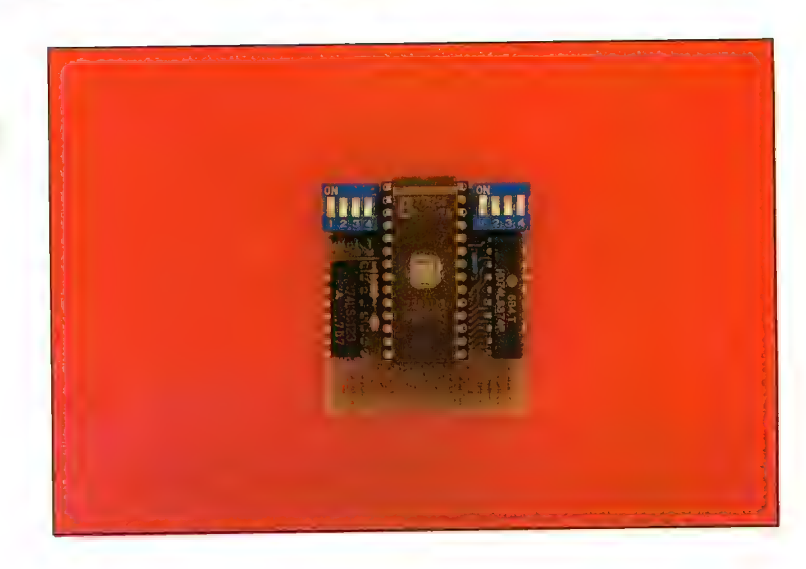
```
 58:
 59:  /*
 60:      NEW_COOKIE:  trägt neuen Cookie in den Jar
 61:                   ein.
 62:
 63:      Achtung !
 64:      Der Fall des bereits vollen Jars wird hier
 65:      noch nicht abgefangen; es muss dann ent-
 66:      sprechend Speicher alloziert, und die ge-
 67:      samte Tabelle umkopiert werden.
 68:  */
 69:
 70:  boolean new_cookie (COOKIE *entry)
 71:  {
 72:      int act_row = 0;
 73:      long old_stack, *cookiejar;
 74:
 75:
 76:      /*  Zeiger auf CookieJar holen  */
 77:
 78:      old_stack = Super (0L);
 79:      cookiejar = *((long **) 0x5a0L);
 80:      Super ((void *) old_stack);
 81:
 82:
 83:      /*  Ist der Jar überhaupt vorhanden ? */
 84:
 85:      if (!cookiejar)
 86:          return (FALSE);
 87:
 88:
 89:      /*  NULL-Cookie suchen  */
 90:      while (cookiejar[0])
 91:      {
 92:          ++act_row;
 93:          cookiejar = &(cookiejar[2]);
 94:      }
 95:
 96:
 97:      /*
 98:          der neue Cookie kann nur eingetragen
 99:          werden, wenn der CookieJar noch aus-
100:          reichend Platz bietet...
101:      */
102:
103:      if (act_row < cookiejar[1])
104:      {
105:          /*  NULL-Cookie weiterschieben  */
106:          cookiejar[2] = cookiejar[0];
107:          cookiejar[3] = cookiejar[1];
108:
109:          /*  neuen Cookie eintragen  */
110:          strncpy ((char *)cookiejar,
111:                   entry->cookie_id, 4);
112:
113:          cookiejar[1] = entry->cookie_value;
114:
115:          return (TRUE);
116:      }
117:
118:      return (FALSE);
119:
120:  }  /*  new_cookie  */
121:
122:
123:
124:  /*
125:      GET_COOKIE:  fragt den Wert eines Cookies
126:                   ab. Als Parameter werden dabei
127:                   die ID des zu suchenden Cookies
128:                   sowie ein Zeiger auf den ge-
129:                   fundenen Wert übergeben.
130:
131:                   Die Routine liefert den Wert
132:                   FALSE falls der angegebene
133:                   Cookie nicht existiert...
134:  */
135:
136:  boolean get_cookie (char *cookie, long *value)
137:  {
138:      long old_stack;
139:      long *cookiejar;
140:
141:
142:      /*  Zeiger auf Cookie Jar holen  */
143:
144:      old_stack = Super (0L);
145:      cookiejar = *((long **) 0x5a0L);
146:      Super ((void *) old_stack);
147:
148:
149:      /*  Ist der Jar überhaupt vorhanden ?  */
150:
151:      if (!cookiejar)
152:          return (FALSE);
153:
154:      do
155:      {
156:          if (!strncmp ((char *)cookiejar,
157:              cookie, 4))
158:          {
159:
160:              /*  nur eintragen, wenn kein
161:                  NULL-Zeiger
162:              */
163:
164:              if (value)
165:              {
166:                  *value = cookiejar[1];
167:                  return (TRUE);
168:              }
169:          }
170:
171:          else
172:          /*  nächsten Cookie nehmen  */
173:          cookiejar = &(cookiejar[2]);
174:
175:      } while (cookiejar[0]);  /* NULL-Cookie ? */
176:
177:      return (FALSE);
178:
179:  }  /*  get_cookie  */
180:
181:
182:
183:  /*
184:      REMOVE_COOKIE:  entfernt den mit 'cookie_id'
185:                      bezeichneten Cookie aus dem
186:                      Jar.
187:  */
188:
189:  void remove_cookie (char *cookie_id)
190:  {
191:      long old_stack, *cookiejar;
192:
193:
194:      /*  Zeiger auf Cookie Jar holen  */
195:
196:      old_stack = Super (0L);
197:      cookiejar = *((long **) 0x5a0L);
198:      Super ((void *) old_stack);
199:
200:
201:      /*  Ist der Jar überhaupt vorhanden ?  */
202:
203:      if (!cookiejar)
204:          return;
205:
206:      /*  suche den zu löschenden Cookie in
207:          der Tabelle...
208:      */
209:
210:      while ((cookiejar[0]) && (strncmp((char *)
211:              cookiejar, cookie_id, 4)))
212:
213:              cookiejar = &(cookiejar[2]);
214:
215:
216:      /*  wurde der gewünschte Cookie gefunden,
217:          oder ist bereits das Ende der Tabelle
218:          (NULL-Cookie) erreicht ?
219:      */
220:
221:      if (cookiejar[0])
222:      {
223:          /*  der Cookie wurde gefunden !  */
224:
225:          do
226:          {
227:              /*  jetzt werden die übrigen Cookies
228:                  einfach eine Position nach oben
229:                  gezogen...
230:              */
231:
```

```
232:            cookiejar[0] = cookiejar[2];
233:            cookiejar[1] = cookiejar[3];
234:
235:            cookiejar = &(cookiejar[2]);
236:
237:        } while (cookiejar[0]);
238:    }
239:
240: }  /*  remove_cookie  */
241:
242:
243:
244: /*
245:     MOVE_COOKIEJAR:  verschiebt (kompletten) Jar
246:                      an eine neue Speicherstelle.
247:                      Als Parameter werden die
248:                      neue Adresse des Jars sowie
249:                      seine Grösse, d.h. die
250:                      Anzahl der in ihn hinein-
251:                      passenden Cookies übergeben.
252: */
253:
254: void move_cookiejar (long *dest, long size)
255: {
256:     long old_stack, *cookiejar, *dest_cop;
257:
258:     /*  Zeiger auf Cookie Jar holen  */
259:
260:     old_stack = Super (0L);
261:     cookiejar = *((long **) 0x5a0L);
262:
263:     /*  Alte Adresse merken  */
264:     dest_cop = dest;
265:
266:     /*  Ist der Jar überhaupt vorhanden ?  */
267:
268:     if (!cookiejar)
269:        return;
270:
271:
272:     do
273:     {
274:         /*  Cookie's kopieren  */
275:         dest[0] = cookiejar[0];
276:         dest[1] = cookiejar[1];
277:
278:         dest      = &(dest[2]);
279:         cookiejar = &(cookiejar[2]);
280:
281:     }while (cookiejar[0]);
282:
283:     /*  NULL-Cookie und seinen neuen (?) Wert
284:         eintragen...
285:     */
286:     dest[0] = cookiejar[0];
287:     dest[1] = size;
288:
289:     /*  _p_cookies auf neue Adresse setzen  */
290:
291:     cookiejar  = (long *) 0x5a0L;
292:     *cookiejar = (long) dest_cop;
293:
294:     Super ((void *) old_stack);
295:
296: }  /*  move_cookiejar  */
297:
298:
299:
300: /*
301:     COOKIE_SIZE:  liefert die Grösse des in-
302:                   stallierten Jars, d.h. die
303:                   Anzahl der maximal möglichen
304:                   Einträge.
305: */
306:
307: long cookie_size (void)
308: {
309:     long old_stack, *cookiejar;
310:
311:
312:     /*  Zeiger auf Cookie Jar holen  */
313:
314:     old_stack = Super (0L);
315:     cookiejar = *((long **) 0x5a0L);
316:     Super ((void *) old_stack);
317:
318:
319:     /*  Ist der Jar überhaupt vorhanden ?  */
320:
321:     if (!cookiejar)
322:         return (0L);
323:
324:     /*  suche den NULL-Cookie  */
325:     while (cookiejar[0])
326:         cookiejar = &(cookiejar[2]);
327:
328:     return (cookiejar[1]);
329:
330: }  /*  cookie_size  */
331:
332:
333:
334: /*
335:     PRINT_COOKIEJAR:  gibt den Inhalt des Jars
336:                       auf die Standard-Ausgabe
337:                       aus.
338: */
339:
340: void print_cookiejar (void)
341: {
342:     long old_stack, *cookiejar;
343:
344:
345:     /*  Zeiger auf Cookie Jar holen  */
346:
347:     old_stack = Super (0L);
348:     cookiejar = *((long **) 0x5a0L);
349:     Super ((void *) old_stack);
350:
351:     /*  Ist der Jar überhaupt vorhanden ?  */
352:
353:     if (!cookiejar)
354:         return;
355:
356:     do
357:     {
358:         /*  Jar auf 'stdout' ausgeben  */
359:
360:         printf („%08lx    %08lx\n",
361:         cookiejar[0], cookiejar[1]);
362:
363:         cookiejar = &(cookiejar[2]);
364:
365:     } while (cookiejar[0]);
366:
367: }  /*  print_cookiejar  */
```

Listing 2

Mikro-EPROM-Karte

8-Bit-EPROM am 16-Bit-Bus des ST



Um am 16 Bit breiten ROM-Port des Atari ST EPROMs anzuschließen, benötigte man bisher mindestens zwei dieser 8 Bit breiten Speicher. Wie es mit einem raffinierten Trick möglich ist, doch mit einem EPROM auszukommen und so Geld und Nerven zu sparen, zeigt dieser Artikel. Sie finden hier eine Bauanleitung für die kleinste 16-Bit-EPROM-Karte der Welt, die zusätzlich noch über vier softwaremäßig abfragbare DIP-Schalter verfügt.

16 Bit, bitte !

Beim Atari ST handelt es sich um einen 16-Bit-Computer. Das bedeutet, daß der Mikroprozessor (CPU 68000) immer, wenn er Daten aus dem Speicher liest oder schreibt, auf 16 Daten-Bits (zwei Bytes) gleichzeitig zugreift. Beim Schreib-Lese-Speicher sind daher in jeder RAM-Bank 16 Speicherbausteine parallelgeschaltet. Der Prozessor erhält aus jedem Baustein ein Daten-Bit, wobei er glaubt, einen 16 Bit breiten Speicher auszulesen.

Der Anschluß von EPROMs am ROM-Port des Atari ST erfolgt bei herkömmlichen EPROM-Karten ähnlich. Gebräuchliche EPROMs verfügen über 8 Datenleitungen, und daher ist es nötig, zwei EPROMs parallelzuschalten, die jeweils die Hälfte der Daten enthalten. Beim Brennen der EPROMs werden die Daten so verteilt, daß EPROM 1 die Hi-Bytes, also die oberen 8 Daten-Bits jedes 16-Bit-Wortes, enthält, während EPROM 2 nur mit den Lo-Bytes gefüllt wird.

Auf der EPROM-Karte sind dabei die 8 Datenleitungen von EPROM 1 mit den Datenbusleitungen D8...D15 des Systems verbunden; die von EPROM 2 sind an D0...D7 des Systemdatenbusses angeschlossen. Bei einem Lesezugriff erhält der Prozessor somit auf D8...D15 das Hi-Byte aus EPROM 1 und auf D0...D7 das Lo-Byte aus EPROM 2, zusammen also ein 16 Bit breites Datenwort, und er merkt gar nicht, daß diese Daten aus zwei EPROMs stammen.

Dieses Verfahren ist zwar einfach, doch wer häufig EPROMs brennt, kennt die Nachteile. Erstens müssen immer zwei EPROMs gelöscht, gebrannt und umgesteckt werden, und zweitens ist das Verfahren teuer. So benötigt man z.B. für 32kByte Daten nicht ein 32K-EPROM, sondern zwei 16k-EPROMs, und das geht ins Geld. Wird nur ein EPROM in dieser Weise angeschlossen, kann der Prozessor pro Lesezugriff nur 8 Daten-Bits lesen, und er wird die Daten-Bytes aus dem EPROM mit jeweils 1 Byte Abstand im Adreßraum vorfinden. Es scheint also kein Weg an einem zweiten EPROM vorbeizuführen.

Daten immer der Reihe nach

Eines Tages fragte mich ein Freund, ob es nicht doch irgendwie gehen könnte. „Kann der Prozessor die Hi- und Lo-Bytes nicht {nach} einander aus dem EPROM lesen?“ fragte er mich, und ich sagte nein, niemals. So eine dumme Frage konnte nur von jemandem kommen, der nichts vom Bus-Timing eines Rechners versteht. Zum Glück verstand ich auch nichts davon, und daher habe ich es ausprobiert.

Das Konzept, wie die Datenleitungen gemultiplext werden, ist vereinfacht in Bild 2 dargestellt. Jedesmal, wenn der Prozessor ein Datenwort aus dem EPROM-Bereich lesen will, muß eine Trickschaltung aktiv werden. Diese muß zunächst das Hi-Byte aus dem EPROM lesen (Phase 1). Nach einer kurzen Zeit, die das EPROM zum Bereitstellen der Daten benötigt, muß das Hi-Byte in einem 8 Bit breiten Speicher zwischengespeichert werden (Phase 2a). Danach kann das EPROM angewiesen werden, das Lo-Byte an den Datenbus zu legen (Phase 2b), und dann kann der Prozessor die Daten auslesen. Er erhält dabei das Lo-Byte direkt vom EPROM, während er das Hi-Byte aus dem Zwischenspeicher bekommt. Auf diese Weise kann er seine 16 Daten-Bits lesen, und weiß nichts davon, welch chaotische Zustände zwischenzeitlich geherrscht haben.

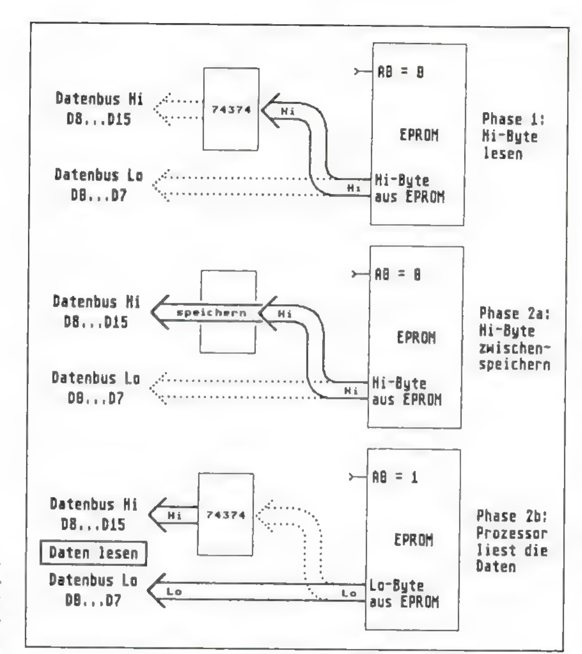


Bild 2: Hi und Lo-Byte werden nacheinander aus dem EPROM gelesen, der Prozessor erhält sie aber parallel.

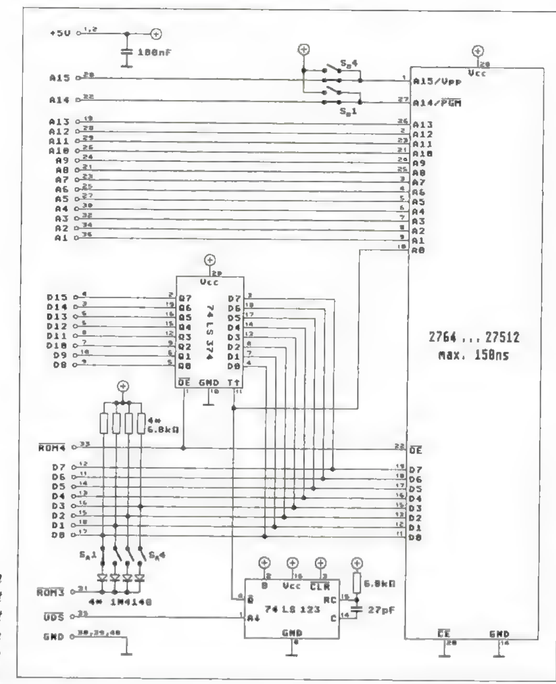


Bild 3: Im Schaltplan sieht man, wie die acht zusätzlichen Datenleitungen gewonnen werden.

Als Zwischenspeicher dient in der Schaltung (Bild 3) ein 8fach-D-Flipflop des Typs 74LS374. Er verfügt über acht Dateneingänge D_N und acht Ausgänge Q_N. Wenn der Zustand seiner Steuerleitung T^+ von 0 nach 1 wechselt, also im Moment der steigenden Flanke, werden die Daten, die in diesem Augenblick an den Eingängen anliegen, intern gespeichert. Eine zweite Steuerleitung, Output Enable $\overline{OE}$, steuert die Ausgänge. Nur, wenn $\overline{OE}$=0 ist, werden die gespeicherten Daten tatsächlich an den Ausgängen zur Verfügung gestellt. Anderenfalls sind die Ausgänge hochohmig, also quasi nicht angeschlossen, und so beeinflussen sie den Systemdatenbus nicht.

Ein zweites IC ist erforderlich, um die Zeit eines Lesezugriffs in zwei Phasen aufzuteilen, um erst das Hi-Byte und dann das Lo-Byte lesen zu können. Verwendet wird ein Monoflop 74LS123. Sein Ausgang $\overline{Q}$ ist normalerweise 1. Wechselt jedoch der Zustand des Eingangs A_* von 1 nach 0, liefert der Ausgang $\overline{Q}$ für eine gewisse Zeit einen 0-Puls. Die Dauer des 0-Pulses wird über einen externen Widerstand und einen Kondensator festgelegt. Es ist größenordnungsmäßig $T_{LOW} \approx 0,7*R*C$; in unserem Fall ist $T_{LOW} \approx 130$ns. Bei der fallenden Flanke an A_* (oder wenigstens später) wechselt $\overline{Q}$ also auf 0, 130ns später wieder auf 1. Während $\overline{Q}$=0 ist (Phase 1), wird dabei das Hi-Byte aus dem EPROM gelesen; wenn anschließend $\overline{Q}$=1 ist (Phase 2), wird das Lo-Byte gelesen.

Das genaue Zusammenspiel der Elemente läßt sich am besten an einem Timing-Diagramm dieser EPROM-Karte erklären. Der zeitliche Verlauf der Bussignale für einen Lesezugriff auf die Mikro-EPROM-Karte ist in einem separaten Textkasten dargestellt und erläutert.

Viele Vielbeiner

Auf der Mikro-EPROM-Karte können Speicher der Typen 2764 bis 27512 betrieben werden. Mit zwei EPROMs im $\overline{ROM4}$-Bereich waren maximal 64kByte zu erreichen (2*27256), und das ist auch mit dieser Karte möglich (1*27512). Eine Ausnutzung der zweiten EPROM-Bank im $\overline{ROM3}$-Bereich hingegen ist mit dieser Karte nicht möglich, während sich auf herkömmliche Weise mit vier 32K-EPROMs eine Speichergröße von 128kByte erreichen ließ.

Da die Pins 1 und 27 bei manchen EPROMs die oberen Adreßleitungen sind, bei anderen aber auf +5V liegen müssen, muß der EPROM-Typ über DIP-Schalter eingestellt werden. Es ist der rechte Satz von vier DIP-Schaltern, markiert mit B.

DIP-Schalter B

Typ	Größe	1	2	3	4
27(C)64	8KB	-	ON	-	ON
27(C)128	16KB	-	ON	-	ON
27(C)256	32KB	ON	-	-	ON
27(C)512	64KB	ON	-	ON	-

Der EPROM-Typ wird über vier DIP-Schalter eingestellt.

Man beachte, daß von den DIP-Schaltern B niemals zwei benachbarte Schalter auf ON stehen dürfen. Anderenfalls kann ein Kurzschluß zwischen Adreßbusleitungen und +5V entstehen.

Da bei der Mikro-EPROM-Karte innerhalb einer Zeit von 300ns *zwei* Lesezugriffe erfolgen, sollten die EPROMs eine

Bild 8a: Das Lese-Timing einer 16-Bit-EPROM-Karte am ROM-Port

Bild 8b: Das Lese-Timing der Mikro-EPROM-Karte

Das Lese-Timing einer 16-Bit EPROM-Karte am ROM-Port

Die Vorgänge, die im einzelnen ablaufen, wenn der Prozessor auf den Speicher zugreift, lassen sich am besten an einem Timing-Diagramm veranschaulichen. (Bereits vor 2000 Jahren verwendeten chinesische Computerexperten Tai-Ming-Diagramme zur Beschreibung von Speicherzugriffen.) Betrachtet wird zunächst eine herkömmliche EPROM-Karte mit zwei EPROMs am ROM-Port. In Bild 8a sind für einen Lesezugriff auf die EPROMs die Zustände von Daten-, Adreß- und Steuerleitungen in Abhängigkeit von der Zeit dargestellt. Aufgetragen sind für Steuerleitungen die Logik-Pegel: oben 1, unten 0. Für Daten- und Adreßleitungen ist aufgetragen, ob sie hochohmig sind oder ob sie definierte Zustände haben, also Daten an ihnen anliegen.

Die Pfeile deuten dabei an, welche Zustandsänderungen andere Veränderungen nach sich ziehen, jeweils mit einer geringen zeitlichen Verzögerung.

Beim dargestellten Lesezugriff setzt der Prozessor zunächst die Adreßleitungen auf das richtige Bit-Muster, um dem Rest des Systems die Adresse der auszulesenden Speicherstelle mitzuteilen. Dann zieht er die Leitungen $\overline{UDS}$ (Upper Data Strobe) und $\overline{LDS}$ (Lower Data Strobe), die mit den $\overline{OE}$-Eingängen (Output Enable) der EPROMs verbunden sind, auf 0, um die Daten anzufordern. Die Memory Management Unit MMU, die den Speicheraufbau organisiert, prüft daraufhin anhand der Adresse, auf welche Speicherbausteine zugegriffen werden soll. Diese erhalten dann von der MMU ein enable-Signal. Für den ROM-Port-Bereich $FA0000 ist es das $\overline{ROM4}$-Signal, es ist mit den $\overline{CE}$-Eingängen (Chip Enable) der EPROMs verbunden; und wenn diese Leitung 0 wird, stellen die EPROMs die angeforderten Daten an den Datenleitungen zur Verfügung. Bis der Prozessor die Daten aber tatsächlich vom Datenbus liest, vergehen ca. 300ns, und das ist auch nötig, da die EPROMs eine gewisse Zugriffszeit benötigen, bis sie die Daten an den Bus legen können. Nach dem Lesen der Daten werden $\overline{UDS}$, $\overline{LDS}$ und $\overline{ROM4}$ wieder 1, und die Datenleitungen der EPROMs werden hochohmig. Der Lesezyklus ist beendet.

Das Lese-Timing der Mikro-EPROM-Karte

Bei der Mikro-EPROM-Karte ist das Bus-Timing (Bild 8b) wesentlich komplizierter, da in der gleichen Zeit nacheinander zwei Zugriffe auf das EPROM stattfinden müssen. Wie üblich setzt der Prozessor zu Beginn des Lesezyklus' die Adreßleitungen, und er zieht $\overline{UDS}$ und $\overline{LDS}$ auf 0. Dies ist das Signal für die Schaltung, aktiv zu werden. Der Monoflop 74LS123, dessen Eingang A ↓ mit $\overline{UDS}$ verbunden ist, beginnt, an seinem Ausgang $\overline{Q}$ einen 0-Puls zu liefern. Da die Adreßleitung A0 des EPROMs mit $\overline{Q}$ verbunden ist, wird sie auch 0, und beim kommenden Lesezugriff wird aus dem EPROM der Inhalt einer gerade Adresse gelesen werden, also das Hi-Byte.

Die MMU legt als nächstes die $\overline{ROM4}$-Leitung auf 0, wodurch sowohl der Zwischenspeicher 74LS374 als auch das EPROM eine 0 am $\overline{OE}$-Eingang erhalten und ihre Daten an den Bus legen. Die Daten des 74LS374 sind irrelevant, es handelt sich um das Hi-Byte des vorigen Lesezugriffs, das noch gespeichert ist. Das EPROM hingegen liefert das aktuelle Hi-Byte, da noch A0=0 ist. Für den Systemdatenbus ist es auch irrelevant, schließlich ist es das Hi-Byte, und bisher liegt es am Lo-Teil des Datenbusses (D0...D7) an, doch es wird vom Prozessor noch nicht gelesen. Viel wichtiger ist, daß die 8 Bits des Hi-Bytes auch an den 8 Eingängen des 74LS374 anliegen. In dem Augenblick nämlich, in dem der 0-Puls des 74LS123 zu Ende ist, werden diese 8 Bits im 74LS374 gespeichert, da ein Ende des 0-Pulses von $\overline{Q}$ mit einer steigenden Flanke an T ↑ des 74LS374 identisch ist.

Das Hi-Byte aus dem EPROM ist also da, wo es hingehört. Es ist im 74LS374 gespeichert, und da $\overline{ROM4}$ und somit $\overline{OE}$ noch 0 ist, liegt es auch tatsächlich am Systemdatenbus D8...D15 an. Auch für das Lo-Byte ist gesorgt, denn mit $\overline{Q}$ hat auch die Adreßleitung A0 des EPROMs wieder auf 1 gewechselt, und daher stellt das EPROM nach einiger Zeit das Lo-Byte an seinen Datenleitungen bereit. Das Bit-Muster der Datenleitungen wechselt also von dem des Hi-Bytes auf das des Lo-Bytes.

Da nun Hi- und Lo-Byte am Systemdatenbus anliegen, kann der Prozessor das 16-Bit-Datenwort vom Bus lesen. Dies geschieht übrigens zu einem festen Zeitpunkt, der unabhängig davon ist, wie lange die Daten schon bereitstehen. Nach dem Lesen geht $\overline{ROM4}$ wieder auf 1, und die Ausgänge des 74LS374 und des EPROMs werden hochohmig, der Lesezyklus ist beendet.

Wer nun argumentiert, das könne doch gar nicht funktionieren, hat Recht. Dem EPROM bleibt eigentlich viel zu wenig Zeit, um in weniger als 300ns auf zwei Bytes zuzugreifen. Selbst bei den schnellsten EPROMs mit nur 150ns Zugriffszeit wäre man erst hart an der Grenze. Tatsache ist jedoch, daß es funktioniert, und zwar sogar mit 250ns-EPROMs bestimmter Hersteller. Der Grund dafür ist, daß die angegebene Zugriffszeit eines EPROMs die Zeit „$\overline{CE}$ to output" ist. Da in der Schaltung aber die Chip Enable Leitung $\overline{CE}$ permanent auf 0 liegt, geht der Zugriff schneller. Relevant für das Lesen sind dann die Verzögerungszeiten von einem 0-Wechsel der $\overline{OE}$-Leitung bis zum Bereitstehen der Daten ($\overline{OE}$ to output) beziehungsweise die Zeit vom Wechsel der Adreßleitung A0 bis zum Wechsel der Daten (A0 to data change). Zumindest die Zeit „$\overline{OE}$ to output" liegt bei allen EPROMs deutlich unter der $\overline{CE}$-Zugriffszeit, garantierte Werte liegen bei 40-65ns. Außerdem werden die Spezifikationen der Speicher stets etwas übererfüllt, die EPROMs sind daher immer schneller, als angegeben ist. Trotzdem sollte man nur EPROMs mit höchstens 150ns Zugriffszeit verwenden.

Betrachtet wurden bisher Lesezugriffe, bei denen der Prozessor ein 16-Bit-Datenwort liest. Die CPU 68000 ist jedoch auch in der Lage, auf einzelne Bytes zuzugreifen. Bei einem Zugriff auf eine gerade Adresse, also auf ein Hi-Byte, verhält sich die Mikro-EPROM-Karte wie bei einem Wortzugriff. Lediglich der Prozessor liest vom Datenbus nur das Hi-Byte, während er das Lo-Byte ignoriert. Greift der Prozessor hingegen auf eine ungerade Adresse zu, also auf ein Lo-Byte, so benutzt er dabei zur Anforderung der Daten nur die $\overline{LDS}$- und nicht die $\overline{UDS}$-Leitung, so daß der Monoflop nicht getriggert wird. Die Folge ist, daß die Mikro-EPROM-Karte sich nun wie eine gewöhnliche EPROM-Karte verhält und das Lo-Byte aus dem EPROM korrekt an den Datenbus legt.

Zugriffszeit von höchstens 150ns besitzen. Diese Zeiten werden fast ausschließlich von EPROMs in der CMOS-Version (27Cxxx) erreicht, die zum Teil schon preiswerter als die Standardtypen sind.

Beim Brennen des EPROMs werden die Daten hintereinander ins EPROM geschrieben. Eine Aufteilung nach geraden und ungeraden Adressen (Hi- und Lo-Bytes) ist nicht erforderlich. Trotzdem verhält sich die Mikro-EPROM-Karte exakt wie eine „echte" 16-Bit-EPROM-Karte.

Die Karte wurde auf den Atari-Modellen 260ST, 520ST, 1040ST, Mega ST und auf dem neuen 1040STE erfolgreich getestet und - man staune - sie läuft sogar auf dem neuen Atari TT030 einwandfrei.

Schalten und walten

Bei dem zweiten Satz von vier DIP-Schaltern, markiert mit A, handelt es sich um softwaremäßig abfragbare Schalter, die z.B. für Konfigurationseinstellungen der Software im EPROM verwendet werden können.

Abgefragt werden sie, indem der Inhalt der Speicherstelle $FB0001 ausgelesen wird. Die Daten-Bits D3...D0 entsprechen dabei den Zuständen der Schalter $S_A4...S_A1$. Steht ein Schalter auf ON, hat das entsprechende Bit den Wert 0.

Realisiert wurde der Anschluß der Schalter nicht über einen Peripheriebaustein, sondern mit einem zweiten Hardware-Trick. Entsprechend dem Schaltplan sind die Datenleitungen über Schalter und Dioden mit der $\overline{ROM3}$-Leitung verbunden. Während $\overline{ROM3}$=1 ist, sperren die Dioden, und die Datenleitungen werden nicht beeinflußt. Wird aber bei einer Abfrage der Speicherstelle $FB0001 die $\overline{ROM3}$-Leitung 0, wird jede Datenleitung, deren Schalter geschlossen ist, über die Diode nach Masse gezogen. Der Prozessor liest auf diesen Leitungen somit den Wert 0 aus. Für alle Datenleitungen mit offenem Schalter hingegen liest der Prozessor den Wert 1 aus, da die 6,8kΩ-Pullup-Widerstände die Leitungen nach +5V ziehen.

Solo für Flötkolben

Davon, die Platine der Mikro-EPROM-Karte selbst herzustellen, kann ich nur abraten. Die Leiterbahnen dieser Miniatur-Platine (Bild 5) sind überwiegend nur 0,25mm breit, und auch eine großzügigere räumliche Anordnung der Bauteile hätte an der engen Leiterbahnführung, speziell im Bereich der Kontaktstreifen, wenig geändert. Wenn eine nicht-durchkontaktierte Platine gefertigt wird, müssen außerdem ca. 30 Durchkontaktierungen durch 0,5mm-Löcher hergestellt werden, und die Bauteile müssen auf Platinenober- und -unterseite angelötet werden.

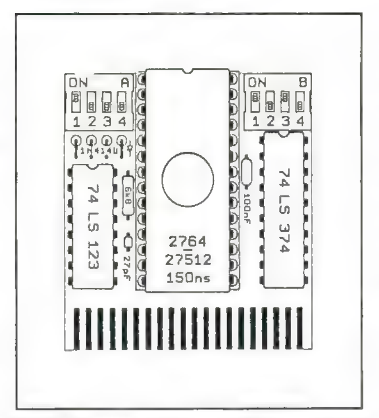


Bild 4: Die enge Bestückung der Platine erfordert

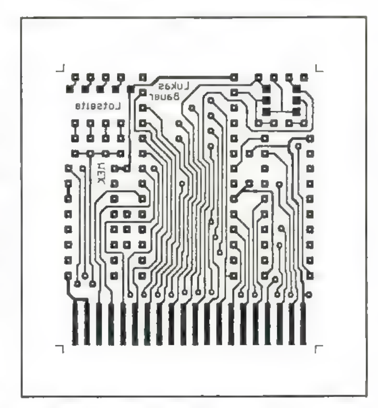


Bild 5a: Platinen-Layout - Lötseite

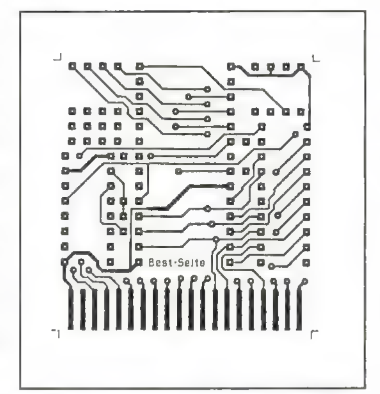


Bild 5b: Platinen-Layout - Bestückungsseite

Wenn hingegen eine industriell hergestellte, durchkontaktierte Platine verwendet wird, dürfte es mit ein wenig Sorgfalt jedem möglich sein, die Platine zu löten. Leider ist die Herstellung durchkontaktierter Platinen bei Einzelstücken sündhaft teuer. Ich bin daher dazu bereit, die Herstellung und den Vertrieb der Leerplatinen zu übernehmen, wenn sich so viele Interessenten bei mir melden, daß es lohnt, eine Kleinserie fertigen zu lassen. Der Preis wird voraussichtlich 20,- DM für die Leerplatine und 40,- DM für das getestete Fertiggerät betragen, inclusive Porto und Verpackung. Allein der Verzicht auf das beidseitige Anlöten der Bauteile wäre mir persönlich immer ein paar Mark Mehrkosten wert.

Achten sollte man beim Aufbau wie immer auf die korrekte Ausrichtung der ICs (Kerbe). Die Werte des 6,8kΩ-Widerstandes und des 27pF-Kondensators sind exakt einzuhalten, da sie die Zeitkonstante des Monoflops festlegen und diese recht zeitkritisch ist. Wer versucht, langsamere EPROMs auf der Karte zu betreiben, und das sollte niemand tun, muß eventuell den Wert des Widerstandes leicht verändern, denn Streuungen bei den Monoflops 74LS123 unterschiedlicher Hersteller führen dazu, daß die Zeitkonstante etwas von den vorgesehenen 130ns abweicht. Bei Verwendung von 150ns-EPROMs stellt dies jedoch kein Problem dar.

Die Dioden werden stehend eingelötet, die Seite mit dem dicken Ring (Kathode) weist von den DIP-Schaltern weg, ist also oben, wenn man sich nach dem abgebildeten Bestückungsplan (Bild 4) richtet. Bei den vier 6,8kΩ-Widerständen handelt es sich um ein single-in-line-Widerstands-Array (Bild 6), bei dem vier Widerstände in einem fünfbeinigen Flachgehäuse untergebracht sind. Da man es selten bekommt, kann man auch ein 8fach-Array mit der Zange kürzen. Es wird als letztes Bauteil auf der Platinenunterseite auf fünf Leiterflächen aufgelötet, wobei der mit einem Punkt markierte gemeinsame Anschluß des SIL-Arrays zur Platinenmitte weist.

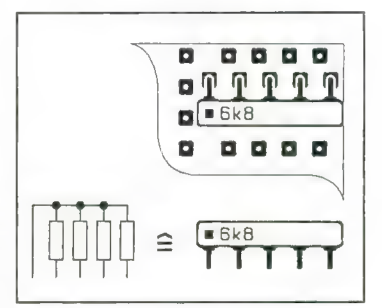


Bild 6: Das SIL-Array wird auf der Platinenunterseite auf fünf Kontaktflächen aufgelötet.

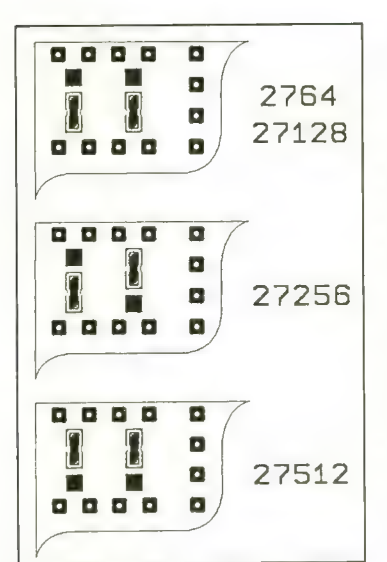


Bild 7: Der EPROM-Typ kann auch fest eingestellt werden. Ein wenig Lötzinn ersetzt die teuren DIP-Schalter.

Für jeden Geldbeutel

Die Bauteile kosten ohne Platine und EPROM ca. 10,- DM. Wer auf die auslesbaren DIP-Schalter (A) verzichten will, kann zusätzlich das SIL-Array und die Dioden weglassen. Auch auf die DIP-Schalter zur Einstellung des EPROM-Typs (B) kann man eventuell verzichten, wenn man die nötigen Verbindungen zwischen den Kontaktflächen auf der Platinenrückseite gemäß Bild 7 mit etwas Lötzinn herstellt. Dies ist jedoch nur empfehlenswert, wenn man ausschließlich mit einem EPROM-Typ arbeitet.

Wer dann noch sparsam und löterfahren genug ist, die IC-Sockel wegzurationalisieren, der zahlt für die Bauteile keine drei Mark mehr, plus Platine und EPROM. Nur dem EPROM sollte man seinen Sockel lassen, das macht sich wirklich besser.

Beim Kauf der Bauteile sollte man im übrigen die enge Bauteileanordung vor Augen haben und darauf achten, daß speziell der Widerstand und die Kondensatoren nicht zu groß sind.

Bezugsquelle:
Lukas Bauer
Goethestraße 30
1000 Berlin 45

Stückliste

1 IC	74LS123
1 IC	74LS374
4 Dioden	1N4148 o.ä.
1 Kondensator	100nF, RM 5mm
1 Kondensator	27pF, RM 2,5mm
1 Widerstand	6,8kΩ
1 SIL-Array	4*6,8kΩ
2	4fach-DIP-Schalter
1	EPROM-Fassung,
28pol.	
1	IC-Fassung, 20pol.
1	IC-Fassung, 16pol.
Platine, EPROM	

PUBLIC DOMAIN PAKETE

1 Graphik

Dieses Paket verwandelt Ihren Atari in eine hochwertige Graphikworkstation. Auf 5 Disketten bieten wir die besten PD-Graphik und Malprogramme. Zum Beispiel

Spezial Paint, The Manipulator, Butterfly Artist, Mistral, Atari Image Manager, Little Painter, Mandelbrot, Turn PIC, GeoScape 3D, Appcher Convert, CAD2, Showfim.

Paket PJ1: 34,90 DM

2 Drucker

Werden Sie zum zweiten Gutenberg mit unseren PD- Druckprogrammen. Ob Poster, Briefköple, oder stilvolle Disklabels. Dieses Paket hat's in sich! Zum Beispiel

The Printing Press, Printing Press Clipart, Idealist, OutPut, Formular, Brief, Barrell, Quickdruck KX P1124, DeltaX, Treiber für IQ800, STX80, P6/7, LC-10 Color u.s.w.

Paket PJ2: 34,90 DM

3 Midi

Sequenzer laden, AMP auf 10 stellen, Cubase◆, Cubeat◆, TwentyFour◆ oder Twelve◆ laden und mit unseren 5 Disketten PD-MIDI Songs ablahren! Zum Beispiel:

Manner - H.Gronemaier, Riders in the Storm - The Doors, Triller - M.Jackson, In the Air tonight - Phil Collins, One Moment in Time, Goldfinger, Crockett's Theme, America, Ghostbusters u.s.w.

Paket PJ3: 34,90 DM

neu Midi 2

Nach dem großen Interesse an unserem ersten Midi-Paket, haben wir uns entschlossen, ein zweites Midi Paket zusammenzustellen. Dabei haben wir uns bemüht nur wirklich gute Stücke aufzunehmen. Hier also 5 Disketten gefüllt mit den neuesten und besten PD MIDI-Songs. Sie werden staunen mit wieviel Perfektionismus einige Stücke eingespielt wurden.

Paket PJ16: 49,90 DM

5 Spiele

Hier bieten wir Ihnen je 5 Disketten mit PD-Spielen quer durch alle Genres

Farbe: Biker, Greece, Stones, Puzzle, Pente, Tunnel Vision, Adel, Chrome, Droid III u.s.w. Monochrom: Bouncing Boubles, Crystal Cave, Esprit Demo (4 Levels spielbar), Braingame, Rat der Weisen, Steins, Swapper u.s.w.

Paket PJ5a: 34,90 DM
Paket PJ5b: 34,90 DM

6 PD-Fonts

Wer mit Signum oder Script arbeitet, der sollte sich diese Pakete zulegen. Jedes Paket enthält 100 P.D.-Zeichensätze. Jeder Zeichensatz liegt als File für 9-Nadeldrucker, 24-Nadeldrucker und Laserdrucker vor.

Paket PJ6a: 49,90 DM
Paket PJ6b: 49,90 DM

7 Disk

Erste Hilfe für den gestressten Disk- /Harddiskbenutzer: 5 Disketten, die vom Harddiskchecker bis zum BackUp alle nötigen PD Programme enthalten. Zum Beispiel

Copy III, Hypercopy, Bitte ein Bit, VDisk, Intram RAM Disk, Fileselect, MakeDisk, Survey, ARC Crunch, DARC, SHAR, UUS, ZOO, Packer, Turtle, Corona Shell, Flexdesk, HD Shell, SM-Shell, Best Shell, Thy Shell, Greifer u.s.w.

Paket PJ7: 34,90 DM

8 Clipart

DTP ohne Bilder ist wie GEM ohne Maus. Hier sind 5 Disketten mit **neuen** hochwertigen Graphiken im PAC- Format



Paket PJ8: 34,90 DM

9 Vector/IMG

Dieses Paket enthält 5 Disketten mit PD-Graphiken im IMG- und Metafile-Vector-Format zum Einsatz unter DTP



Paket PJ9: 34,90 DM

neu Clipart 2

Paket 14a enthält 5 Disketten gefüllt mit hochwertigen Grafiken im PAC-Format (Bisher in keiner Sammlung). Auf Paket 14b befinden sich auf 10 Disketten Grafiken im IMG-Format (bisher in keiner Sammlung). Beide Pakete enhalten Archivblätter zum schnellen Auffinden der Grafiken

Paket PJ14a: 34,90 DM
Paket PJ14b: 49,90 DM

neu Neuheiten

In diesem Paket haben wir auf 6 Disks die neueste PD, die unseres Wissen bisher in keiner großen Serie erschienen ist zusammengestellt. Z.B.

Postkart - Programm zum Erstellen luxoriöser Postkarten. Verein - Umfangreiche Vereinsverwaltung. ARC 6.02, THARC 1.3B, Les... druckt Stundenpläne, Volumina - berechnet Flächeninhalte, Schulmanager - integriertes ...gramm mit Spickzettel, Entschuldigung u.s.w. Puzzle Willi - tolles Farbspiel u.v.m.

Paket PJ15: 39,90 DM

Bitline

W. Wohlfahrstätter und J. Ohst EDV GbR

Jutta Ohst ■ Nelkenstraße 2 ■ 4053 Jüchen 2 ■ Tel.: 02164/7898

Hardware ✧ Software

CameoST

Daß auch professionelle Software nicht zwangsläufig teuer sein muß, beweisen wir mit unserem CameoST einem wahren Multitalent.

CameoST, das Musikarchiv

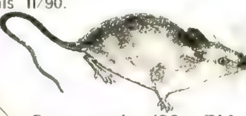
CameoST ist eine Datenbank für CDs, LPs und MCs. CameoST verfügt über flexible Suchfunktionen, die nicht nur alle Beethoveneinspielungen oder alle Versionen von 'Just the two of us', sondern auch alle verliehenen LPs in Sekundenschnelle ermittelt. Fest ins CameoST Konzept eingebunden ist eine Adresskartei, in der Sie allen gespeicherten Personen auch beliebige Eigenschaftskennzeichen zuweisen können. Eine weitere Funktion ist der eingebaute Notizblock. **Einen ausführlichen Testbericht entnehmen Sie bitte der Zeitschrift 'PD Journal' 11/90**

Superpreis: 59,- DM

Cordless Mouse

Die innovative Infrarottechnik, die das lästige Mauskabel unnötig macht und ein ergonomisches Design stellen das Original weit in den Schatten. Dabei ist die Installation so einfach wie bei der alten Maus und Kompatibilitätsprobleme treten auch nicht auf.

Näheres erfahren Sie in dem Testbericht der 'TOS 10/90' bzw. 'PD-Journals' 11/90.



Superpreis: 198,- DM

Mato-Vektor Zeichensätze

Wir bieten Ihnen Vector-Fonts aus eigener Herstellung für Calamus®✧. Damit Sie eine reichhaltige Auswahl an Schriften zu einem wirklich günstigen Preis erhalten, haben wir Schriftenpakete für Sie zusammengestellt. Ein Paket enthält bis zu 26 Schriften. Fordern Sie unsere Übersicht an. Diese Anzeige wurde übrigens mit dem Vektorfont Serif gesetzt und belichtet.

DTP-Schriften
Paket nur: **99,-**

Showtime^Pro

Dieses graphische Präsentation- und Informationsprogramm reizt die Möglichkeiten Ihres ST voll aus. Allein 40 absolut flimmerfreie, saubere Umblendalgorithmen, GOTO, REPEAT UNTIL- Strukturen, Multitasking-Sound, Vernetzung vieler Ataris mittels MIDI und die perfekte Benutzerführung machen dieses Programm zum Novum. Innerhalb kürzester Zeit erstellen Sie perfekte Diashows oder nutzen durch die beliebigen Sprungmöglichkeiten auf Tastendruck das Programm zum Aufbau eines Informations- bzw. Lernssystem.

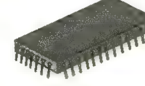
Interesse?

Für weitere Informationen stehen wir Ihnen gerne zur Verfügung.

Einführungspreis: 99,- DM

Speicher

Speichererweiterung für AtariST Computer von 512 KB auf 1 MB oder von 512KB, 1MB auf 2,5/4 MB. Die Speichererweiterung ist voll steckbar und für alle Computertypen (außer MegaST), deren MMU gesockelt ist geeignet. Die große Aufrüstung kann in zwei Stufen erfolgen (2,5/4MB).



1 MB	198,- DM
2 MB	449,- DM
4 MB	698,- DM

Auf Wunsch übernehmen wir auch den Einbau.

Einbau incl. Porto 50,- DM

Scanner

Bildverarbeitung für jedermann?

Ja. Denn jetzt gibt es Repro Studio junior. Dazu gehört nicht nur ein Logitech Handscanner mit 100-400 dpi, 32 Graustufen, 3 Raster, 105 mm Rasterbreite, sondern auch das professielle Bildverarbeitungsprogramm Repro Studio ST junior. Das alles erhalten Sie zu einem Preis, den Sie bisher nur für einen Scanner bezahlt hätten.

ReprostudioST junior: 598,- DM incl. Scanner

Weihnachts Pakete

Sie suchen ein schönes Geschenk für Ihre(n) computerbesessene(n) Freund, Freundin, Partner(in), Sohn, Tochter u.s.w. – wir haben das Richtige für Sie zusammengestellt und weihnachtlich verpackt. Auf Wunsch liefern wir auch per Express (Versandkosten + 8,- DM).



Datenbank Textverarbeitung

That's Write	328,- DM
I.D.A.	358,- DM
Signum!2	398,- DM
Script 2	278,- DM
Daily Mail	169,- DM
STAD	169,- DM
That's Pixel	139,- DM

Zubehör

10 Disketten 2DD	14,90 DM
50 Disketten	69,00 DM
Monitor Switchbox	49,90 DM
Auto Switchbox	59,90 DM
Druckerkabel 2m	29,90 DM
Scart Kabel	29,90 DM
Verlängerungskabel Festplatte	49,90 DM
Farbband NEC P6	17,90 DM
Farbband Epson LQ 500,800,850	14,90 DM
Farbband Star LC10	11,90 DM

Weihnachts Pakete

5 ausgesuchte Public Domain Disk
5 Leerdisketten
1 Showtime-Demo
1 Mausmatte

Paket W1 nur 59,- DM

100 bunte Endlosetikettenlabels
10 ausgesuchte Public Domain
10 Leerdisketten
1 Showtime-Demo
1 Mausmatte

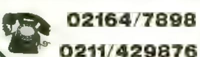
Paket W2 nur 99,- DM

Verschiedenes

AT-Speed	478,- DM
Best 2400L	298,- DM
Best 2448LF	398,- DM
Das DFU-faxmodem incl. Atari-Software	
MultiTerm pro an Modem	158,- DM
Bildschirmtext Programm	
MultiTerm pro an D-BT 03 in Farbe und sw	236,- DM
Showtime-Demo	10,- DM
Cameo-Demo	10,- DM

Versandkosten

Der Versand erfolgt per Nachnahme oder Vorauskasse. Natürlich können Sie alle Produkte auch telefonisch per Nachnahme bestellen. **Die Versandkosten betragen 3,00 DM bei Scheck- oder Barzahlung und 6,00 DM bei Nachnahme.** BTX 0211429876





Werner Wohlfahrtstätter ■ Irenenstraße 76c ■ 4000 Düsseldorf 30 ■ Tel.: 0211/429876

Hallo Freaks,



auch in diesem Monat strömten wieder massig Neuigkeiten in die Redaktion. Hier und da hörte man auch einen Aufschrei à la „Ich kann den Monitor vor lauter Spielen nicht mehr sehen". Oder war's der Wald? Na, egal. Kommen wir zur Sache.

Im letzten Monat angekündigt und nun doch nicht da, ist ein Bericht über Videospiele. Nächsten Monat kommt er aber - versprochen.

Compilations über alles. So scheinen momentan viele Spielefirmen zu denken. So gibt's auch diesen Monat wieder eine zu kaufen. „Sega Master Mix" nennt sich das Päckchen mit den dynamitartigen Spielen. Dem Spaß auf die Sprünge helfen sollen dabei **Super Wonder Boy**, **Turbo Outrun**, **Crack Down**, **Dynamite Düx** und **Thunder Blade**. Da heißt es zugreifen, denn eine solche Gelegenheit wird sich nicht so schnell wieder bieten!



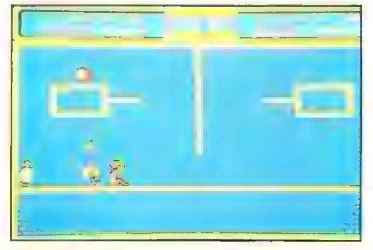
Die Brut der Nacht: Na ja...

Horrorfans werden auch in diesem Monat nicht auf ihre Kosten kommen. Kennen Sie den Film „**Die Brut der Nacht**"? Dann kennen Sie auch schon die Handlung des Spiels. Sie sind Boone, etwas verrückt, und müssen nach Midian, um dort die Brut der Nacht zu retten: eine fantastische Rasse, die schon lange vor der Menschheit auf der Erde war und von den ach so bösen Menschen fast ausgerottet wurde. Das Spiel ist derart langweilig, daß man nur sagen kann: „O Gott, laß dieses Spiel an mir vorüberziehen!"



Gremlins 2: Pünktlich zum Film

Kaum sind die Gremlins wieder im Kino, taucht auch das passende Spiel dazu auf. **Gremlins 2 - The New Batch** nennt sich das Game. Im Spiel sind Sie Billy und müssen Ihre Reise durch Camp Plaza heil überstehen. Hin und wieder hilft auch Gizmo dabei. Natürlich kommen aus allen Ecken und Enden Gremlins hervor, die Ihnen das Leben schwer machen. Typisch. Gremlins 2 ist gut gelungen, wenn es auch im Schwierigkeitsgrad recht hoch angesiedelt ist. Nicht nur für Gremlins-Fans empfehlenswert!



Helter Skelter: Ein neues Plattformspiel

Spielideen scheinen schwierig zu finden zu sein. **Helter Skelter** ist eines der Spiele, das man nicht unbedingt in seinem Diskettenkasten haben muß. Ich sage nur ein Wort: „Plattform". Natürlich, nicht schon wieder, denken Sie. Immerhin hat Audiogenic noch eine nette Variation in das Spiel gebracht. Mit Ihrem Spielball „Billy" (so hieß doch schon der aus Gremlins?) springen Sie durch 80 Levels und müssen sich auf Monster plumpsen lassen. Nur eines ist verwundbar. Treffen Sie aus Versehen das falsche, teilt es sich in zwei Monster auf. Kennen wir das irgendwoher schon? Für Fans von Plattformspielen ist Helter Skelter unentbehrlich, ansonsten vernachlässigbar.



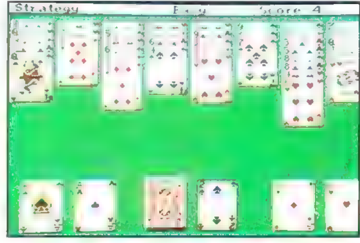
Spiderman: Es wird gekrabbelt

Spider-Man krabbelt wieder munter durch die Gegend. Sein Kollege Supermann hat ja in der Comic-Serie in Amerika inzwischen geheiratet und ist aus dem Spiel. So bietet United Software nun **The amazing Spider-Man** als Spiel an. Die Packung meint: „Schlage Deine Zähne in dieses Spiel, und Du wirst zum Süchtigen". Süchtig: ja. Allerdings süchtig vom Kaffee, der benötigt wird, um Spider-Man ohne Schaden zu überstehen. Sinnloses Herumgekrabbele an allen möglichen Wänden, Schalter, die nicht zu sehen sind, und andere Nettigkeiten machen das Spiel zum Trip durch die Langeweile. Da kann auch der mitgelieferte Original-Comic nicht helfen. Anders gesagt: Das Spiel hält, was der Comic verspricht.

Die definitive Simulation des Panzerkampfes soll alle schießwütigen Spieler zufriedenstellen. Wie man aus einem solchen Thema eine Simulation machen kann, ist die Frage. „Planen Sie Ihre siegreichen Strategien, erteilen Sie Ihrem **Panzer-Platoon** die richtigen Befehle zur richtigen Zeit", heißt es auf der Packung. Erteilen Sie sich selbst den Befehl, solcherlei „Spiele" zu meiden. Auch wenn's nur eine Simulation ist, hätte sich Microprose sicher ein besseres Thema als Panzer aussuchen können!



Tank Platoon: Simulation gut, Thema übel



Hoyle schlägt zu: 28 Spielversionen

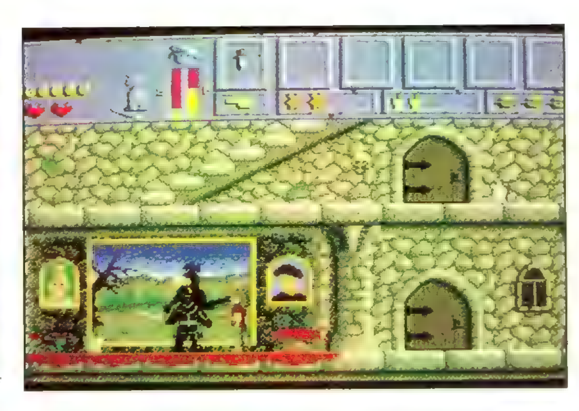
Spielen Sie gerne Solitaire? Haben Sie 80 Mark und wissen nicht, wohin damit? Dann sind Sie mit **Hoyle Official Book Of Games Volume 2 Solitaire** gut bedient. 28 verschiedene Solitaire-Spiele in einem Programm können sich sehen lassen. Sie sollten allerdings englisch sprechen können, denn der Aufkleber „Mit ENGLISCHER Bedienungsanleitung" klebt nicht grundlos auf der Packung. Lassen Sie sich auch nicht durch den Schriftzug „Runs great on laptops!" beeindrucken - selbst wenn das Spielchen auch auf dem Stacy läuft (in Farbe und monochrom), ist der Hinweis wohl eher für PC-Anwender gedacht... Alles in allem ist Solitaire ein recht gelungenes Spiel, das dem Kartenspielfan nur empfohlen werden kann.

MP

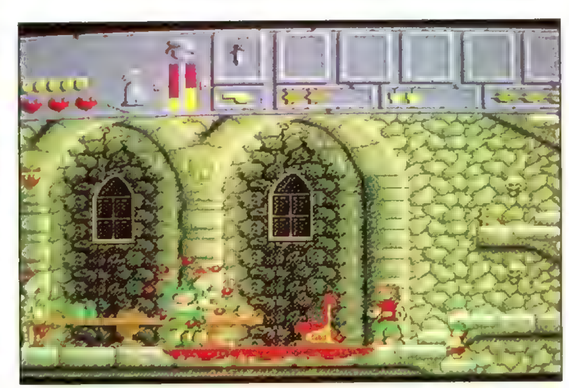
Tom and the Ghost

Endlich mal wieder ein Geist-volles Spiel. Im Action-Adventure „Tom and the Ghost“ wimmelt es nur so von Geistern, und seinen eigenen Grips anstrengen muß man auch. Dem Genre entsprechend ereignet sich die folgende Geschichte in einer schottischen Burg. Der 7jährige Steppke Tom besichtigt mit seiner Mutter zusammen die Rittersäle und Verliese des mittelalterlichen Gemäuers. Unheimliche Dinge haben sich hier zugetragen, wenn man dem Fremdenführer glauben will. Die Spuk-Story bewahrheitet sich nur allzu schnell. Tom sieht gerade noch, wie eine finstere Gestalt seine Mutter packt und beide sich in Luft auflösen. Wer würde Tom so etwas glauben? Wer könnte ihm helfen? Schwupp, da ist er schon, der edle Retter, äh, Ritter. Es ist Sir Arrow, von Beruf Geist. Nach 500 Jahren müßigen Herumspukens kann er sich jetzt einmal nützlich machen. Er bietet dem kleinen Tom seine Hilfe an. Von nun an ist Sir Arrow Kämpfer und Babysitter zugleich. Tom ist nämlich ein recht empfindsamer und verwöhnter Dreikäsehoch. Das einzige, was ihn zur Folgsamkeit bewegt, ist eine Trillerpfeife. Flötet Arrow darauf, kommt Tommi herbei; aber wenn der Bengel sich gerade vor einem weißen Gespenst gruselt, bleibt er trotzdem in seinem Versteck, beispielsweise hinter einer morschen Kerkertür.

Woran es dem Kleinen gerade fehlt, sieht man auf zwei Pegelanzeigen, eine für Energie und eine für seelisches Befinden. Die Kraftreserven lassen sich schnell mit Hamburger-Brötchen und Milch-Shakes auffüllen. Sir Arrow muß nur darauf achten, daß der Kindermagen nicht zuviel davon bekommt. Wenn Tommi Angst hat, sollte er mit einem Teddy beruhigt werden. Ritter Arrow muß sich auch um sein eigenes Wohlergehen kümmern. Gebratenes Geflügel ist für einen Geist wie ihn genau die rechte Kost. Rote Powerdrinks geben ebenfalls Kraft gegen böse Angreifer. Zuerst müssen sich die beiden Helden gegen vergleichsweise harmlose Gegner verteidigen. Zum Glück, denn anfangs ist Sir Arrow nur mit Messer und Streitaxt bewaffnet. Und Tom unterstützt ihn im Kampf nach Kräften mit dem Baseballschläger, denn - boing! - auch das ist eine gute Antigeisterwaffe.



Je weiter das Spiel vorangeht und je tiefer Ritter und Junge in die Gewölbe eindringen, desto mehr wichtige Gegenstände finden sie: Zaubertränke, Feuerzeug, Schlüssel und Waffen wie Morgenstern, Armbrust, Hellebarde, Tränengas und Schwert. All der zusammengetragene Krempel ist in einem speziellen Inventarfenster zu sehen. Über den gerade benötigten Gegenstand wird ein goldener Rahmen manövriert, und der Ritter hält das gewünschte Utensil in der Hand. Jedes hat seinen bestimmten Zweck: das Tränengas schützt beispielsweise nur vor den schwebenden Augen. Ein blauer Zaubertrank verleiht übermenschliche Sprungkraft, das Schwert entscheidet im Duell mit dem schwarzen Ritter über Leben und Tod.



In „Tom and the Ghost“ gibt es schnelle Action und einiges zum Grübeln. Welcher Gegenstand aus dem Inventar mag gegen Vampire helfen? Natürlich der Holzpflock. Welche Schlüssel passen zu welcher Tür? Jede Waffengattung hat unterschiedliche Eigenschaften, was Schnelligkeit und Durchschlagkraft betrifft. Darüber sollte man sich zuvor schlau machen, um im gefährlichsten Moment die geeignete Waffe zu zücken. Und nebenbei lohnt sich ein Blick auf die Energieanzeige beider Charaktere. Wer viel kämpft, muß auch viel futtern. Sobald die Energie ein gewisses Limit unterschreitet, ist die Spielfigur am Ende - und das Spiel auch.

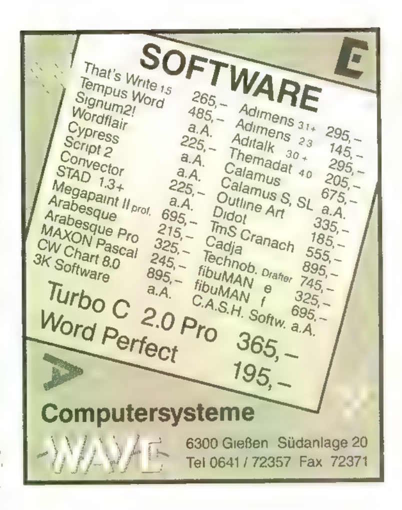
Zur Steuerung ist der vorgesehene Joystick sicherlich das geeignete Instrument. Ritter Arrow läßt sich allerdings auch über den Ziffernblock lenken. Dabei ersetzt die Taste „5“ den Feuerknopf. Komplexere Bewegungen wie ein Schwertstreich werden dann durch Tastenkombinationen aktiviert. Hauptsache, es geht fix. Tempo ist eine hervorragende Eigenschaft des Spiels, was im wesentlichen darauf zurückzuführen ist, daß für die Programmierung ein pfiffiger Turbo-Assembler verwendet wurde. In Sachen Komplexität hat das Geisterspiel ebenfalls einiges zu bieten. Auch die Grafik ist hübsch, niedlich und bunt. Schöne Musik und Soundeffekte, die mal nicht Ohr und Nerven strapazieren, begleiten das Adventure. Wer putzige Action-Spiele mag und gern dabei knobelt, sollte sich „Tom and the Ghost“ unbedingt einmal anschauen.

CBO

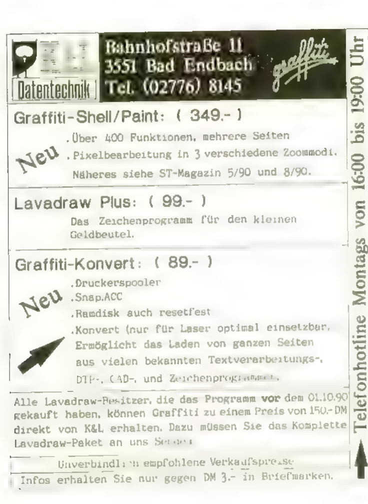




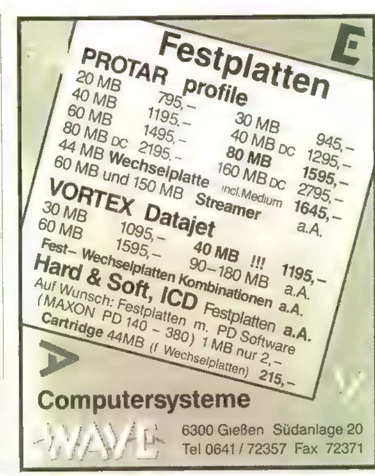




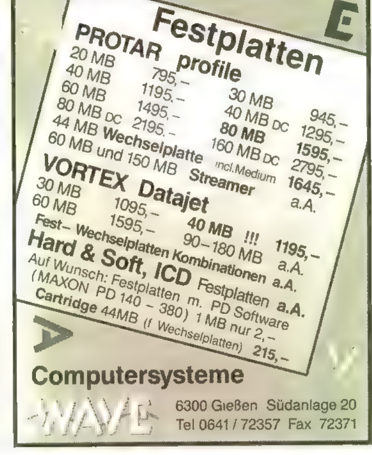
SOFTWARE
That's Write 1.5 265,–
Tempus Word 485,–
Signum2! a.A.
Wordflair 225,–
Cypress a.A.
Script 2 a.A.
Convector 225,–
STAD 1.3+ a.A.
Megapaint II prof. 695,–
Arabesque 215,–
Arabesque Pro 325,–
MAXON Pascal 245,–
CW Chart 8.0 895,–
3K Software a.A.
Adimens 3.1+ 295,–
Adimens 2.3 145,–
Aditalk 3.0+ 295,–
Themadat 4.0 205,–
Calamus 675,–
Calamus S, SL a.A.
Outline Art 335,–
Didot 185,–
TmS Cranach 555,–
Cadja 895,–
Technob. Drafter 745,–
fibuMAN e 325,–
fibuMAN f 695,–
C.A.S.H. Softw. a.A.
Turbo C 2.0 Pro 365,–
Word Perfect 195,–
Computersysteme
WAVE
6300 Gießen Südanlage 20
Tel 0641 / 72357 Fax 72371



Bahnhofstraße 11
3551 Bad Endbach
Tel. (02776) 8145
Datentechnik
graffiti
Graffiti-Shell/Paint: (349.-)
Neu
.Über 400 Funktionen, mehrere Seiten
.Pixelbearbeitung in 3 verschiedene Zoommodi.
Näheres siehe ST-Magazin 5/90 und 8/90.
Lavadraw Plus: (99.-)
Das Zeichenprogramm für den kleinen Geldbeutel.
Graffiti-Konvert: (89.-)
Neu
.Druckerspooler
.Snap.ACC
.Ramdisk auch resetfest
.Konvert (nur für Laser optimal einsetzbar. Ermöglicht das Laden von ganzen Seiten aus vielen bekannten Textverarbeitungs-, DTP-, CAD-, und Zeichenprogrammen.
Alle Lavadraw-Besitzer, die das Programm vor dem 01.10.90 gekauft haben, können Graffiti zu einem Preis von 150.- DM direkt von K&L erhalten. Dazu müssen Sie das Komplette Lavadraw-Paket an uns Senden.
Unverbindlich empfohlene Verkaufspreise
Infos erhalten Sie nur gegen DM 3.- in Briefmarken.
Telefonhotline Montags von 16:00 bis 19:00 Uhr

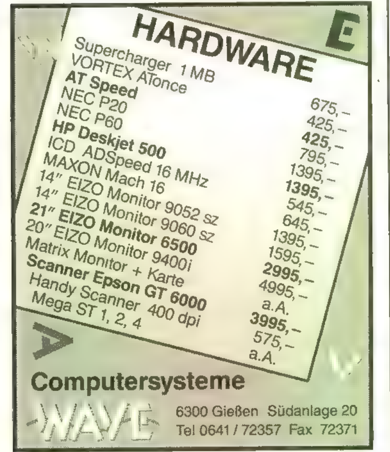


Festplatten
PROTAR profile
20 MB 795,–
40 MB 1195,–
60 MB 1495,–
80 MB DC 2195,–
30 MB 945,–
40 MB DC 1295,–
80 MB 1595,–
160 MB DC 2795,–
44 MB Wechselplatte incl.Medium 1645,–
60 MB und 150 MB Streamer a.A.
VORTEX Datajet
30 MB 1095,–
60 MB 1595,–
40 MB !!! 1195,–
90–180 MB a.A.
Fest– Wechselplatten Kombinationen a.A.
Hard & Soft, ICD Festplatten a.A.
Auf Wunsch: Festplatten m. PD Software
(MAXON PD 140 – 380) 1 MB nur 2,–
Cartridge 44MB (f Wechselplatten) 215,–
Computersysteme
WAVE
6300 Gießen Südanlage 20
Tel 0641 / 72357 Fax 72371

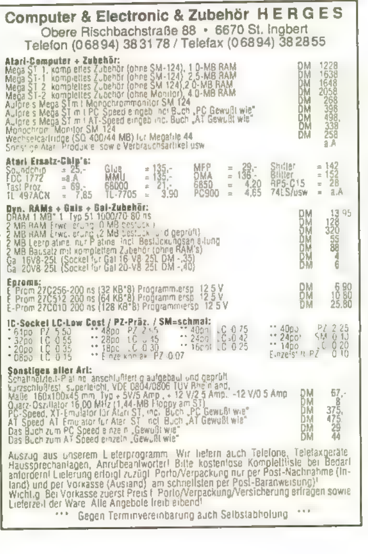




1st WORD plus
Das Standardtextprogramm
Version 2.02, incl. 1st Mail, mit dt. Handbuch und Updateservice. 58 Stück lieferbar!
nur 59,– DM



HARDWARE
Supercharger 1 MB 675,–
VORTEX ATonce 425,–
AT Speed 425,–
NEC P20 795,–
NEC P60 1395,–
HP Deskjet 500 1395,–
ICD ADSpeed 16 MHz 545,–
MAXON Mach 16 645,–
14" EIZO Monitor 9052 sz 1395,–
14" EIZO Monitor 9060 sz 1595,–
21" EIZO Monitor 6500 2995,–
20" EIZO Monitor 9400i 4995,–
Matrix Monitor + Karte a.A.
Scanner Epson GT 6000 3995,–
Handy Scanner 400 dpi 575,–
Mega ST 1, 2, 4 a.A.
Computersysteme
WAVE
6300 Gießen Südanlage 20
Tel 0641 / 72357 Fax 72371



Computer & Electronic & Zubehör HERGES
Obere Rischbachstraße 88 • 6670 St. Ingbert
Telefon (06894) 383178 / Telefax (06894) 382855
Atari-Computer + Zubehör:
Atari Ersatz-Chip's:
Dyn. RAMs + Gals + Gal-Zubehör:
Eproms:
IC-Sockel LC-Low Cost / PZ-Präz. / SM=schmal:
Sonstiges aller Art:
Gegen Terminvereinbarung auch Selbstabholung

Vaxine



Im menschlichen Körper tobt ein permanenter Kampf zwischen eindringenden Viren und dem Immunsystem. Meist gewinnt der Körper leicht die Oberhand und besiegt die Eindringlinge, doch hin und wieder tritt ein stärkerer Virenstamm auf - Mutanten, und die Abwehrmechanismen des Körpers laufen Gefahr, überwältigt zu werden. Geschieht das, benutzen wir künstliche Mittel, um den Körper zu schützen, und mit Vaxine steuern Sie einen der neuesten Abwehrstoffe. Sie fahren durch Ihren Körper in 3D-Ansicht. Vorwärts, rückwärts, links und rechts können Sie sich als Abwehrstoff steuern. Ihre Möglichkeiten bei der Zerstörung der Viren sind beschränkt, aber wirkungsvoll. Jeder Virus, den es im Körper zu bekämpfen gilt, hat eine andere Farbe. Er muß mit Kugeln genau der gleichen Farbe beschossen werden. Das geht folgendermaßen vor sich: Treffen Sie beispielsweise auf einen roten Virus, drücken Sie die rechte Maustaste, bis Ihre Bekämpfungskugeln ebenfalls rot sind. Dann feuern Sie sie auf den Virus ab. Treffen Sie den Virus, ist er vernichtet und kann im Körper keinen Schaden mehr anrichten. Treffen Sie ihn jedoch mit einer falschen Farbe, entstehen Sterne, die innerhalb von 5 Sekunden abgeschossen werden müssen - sonst werden Ihnen Schüsse aus Ihrem sowieso schon begrenzten Vorrat abgezogen.

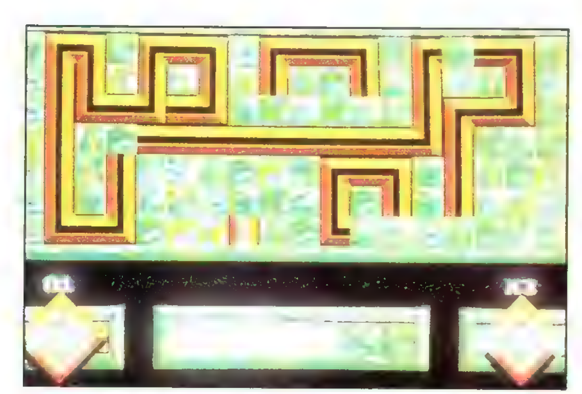
Wenn zwei oder mehr gleichfarbige Viren aufeinandertreffen, bilden sie ein Konsortium und verbinden sich. Wenn sie sich dann auch noch an einer Basis auf dem Boden verankern, kann es schon zu spät für Sie sein, denn so verlieren Sie immer mehr die Kontrolle über den Körper - es ist halt wie im echten Leben. Wenn die Viren nicht rechtzeitig bekämpft werden, breiten sie sich unaufhörlich aus und schwächen die Abwehrkräfte des Körpers.

Vaxine ist unglaublich interessant und, man wagt es kaum zu sagen, endlich einmal eine neue Spielidee. Die Grafik ist exzellent gelungen, man kann sich kaum noch vom Bildschirm losreißen, der irre 3D-Effekte darstellt. Vaxine ist das erste Spiel, das knapp an der „10" für die Grafik gescheitert ist. Die Titelmusik des Spiels ist ebenfalls vom Feinsten, denn sie ist gesampelt und hört sich wirklich wunderbar an. Das Spiel ist dadurch eines der besten, das man zur Zeit spielen kann. Vaxine sollte in keiner Spielesammlung fehlen.

MP



Loopz



Tetris scheint noch immer ein so großer Erfolg zu sein, daß andauernd Spiele auf den Markt kommen, die an das Spielprinzip des Veteranen anknüpfen wollen. So hat Audiogenic jetzt Loopz entwickelt, einen weiteren Tetris-Coloris-Block-Out-Lettrix-wie-auch-immer-Clone. Hier geht es allerdings nicht darum, eine waagerechte oder senkrechte Linie zusammenzusetzen, sondern die Aufgabe besteht darin, ein zusammenhängendes N-Eck zu erzeugen. Je größer das N-Eck ist, desto mehr Punkte gibt's dafür. Zum Zusammensetzen setzt das Programm dem Spieler (zufallsberechnet) Bausteine vor, die er einpassen muß. Da gibt es kleine Kanten, große Kanten, kleine Geraden, große Geraden, ein kleines „s", ein großes „U" und viele andere Bausteine, die natürlich immer dann erscheinen, wenn man sie nicht verwenden kann. Hat man ein N-Eck zusammengesetzt, verschwindet es vom Bildschirm und schafft damit Platz für ein weiteres Gebilde.

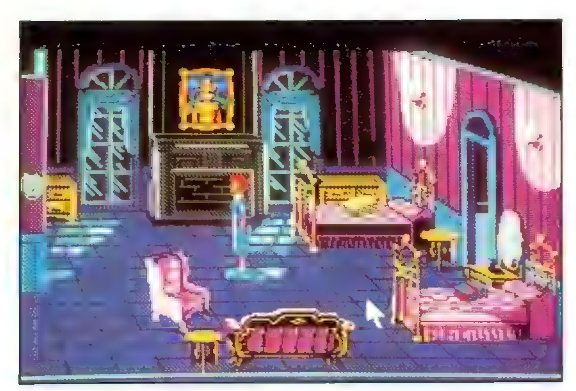
Das Spiel wäre nicht Loopz, wenn man nicht auch noch gegen die Zeit spielen müßte. Zum Ablegen eines Steins hat man nur einige Sekunden Zeit. Wird diese überschritten, gilt der Zug als Fehlversuch. Nach drei Fehlversuchen ist das Spiel beendet. Nun kommt man aber leider auch ab und zu in die Verlegenheit, einen Spielstein dort plaziert zu haben, wo man ihn eigentlich gar nicht ablegen wollte. Er würde immer an der Stelle liegenbleiben, wenn es nicht von Zeit zu Zeit die besondere Funktion gäbe, zusammenhängende Linien vom Spielfeld zu entfernen. Dieser Bonus erscheint jedoch erst ziemlich spät, manchmal ist es schon zu spät, nämlich genau dann, wenn nur noch große Steine erscheinen, die man nicht mehr auf dem Spielfeld plazieren kann.

Loopz ist ein absolut überzeugendes Spiel, das jedem Tetris- und Klax-Fan nur warm ans Herz gelegt werden kann. Endlich mal wieder ein Spiel, das nicht nur die ersten zehn Minuten Spaß macht, sondern auch noch Wochen danach. Wer Denk- und Knobelspiele mag, ist mit Loopz auf jeden Fall gut bedient. Wer übrigens nicht gerne alleine spielt, ist mit Loopz ebenfalls gut beraten, denn das Spiel beinhaltet einen exzellenten Zwei-Spieler-Modus. Dabei legen die beiden Spieler abwechselnd Teile auf den Bildschirm. Das macht mehr Spaß, als man sich vorstellen kann. Wie die Packung des Spiels schon sagt: „Sehr bald werden Sie von Loopz träumen" - so weit ist es zwar noch nicht gekommen, aber das Spiel fesselt.

MP



The Colonel's Bequest



Die Kette von Sierra-Adventures weist einige sehr bekannte Spiele auf, und ein paar davon sind von der Programmiererin Roberta Williams geschrieben, so etwa die komplette King's Quest-Serie. Ihr neuestes Werk, The Colonel's Bequest, ist auf dem Weg, sich einen Platz in den Charts zu sichern. Die Handlung spielt im Jahr 1925. Sie sind Laura Bow, eine junge College-Studentin, und Sie sind eingeladen worden, das abseits gelegene Anwesen des Colonels zu besuchen. Als Sie ankommen, merken Sie sofort, daß etwas nicht stimmen kann mit den Menschen, die sich auf dem Anwesen aufhalten. Sie müssen den Gefahren entkommen, die sich Ihnen überall entgegenstellen und den Killer finden, bevor er oder sie wieder zuschlägt.

Versuchen Sie herauszufinden, wer sich in den Räumen des Anwesens herumtreibt und die Gäste des Colonels umbringt. Entdecken Sie versteckte Räume, belauschen Sie Intrigenpläne und Gespräche ...

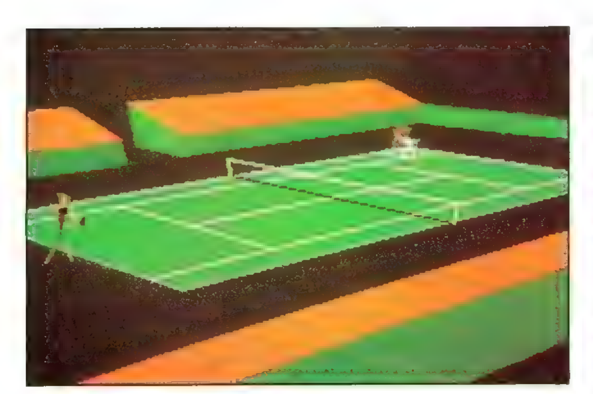
Das neueste Adventure hat einen recht hohen Schwierigkeitsgrad und läßt einen, wie man es gewohnt ist, nach dem ersten Einladen schon nicht mehr los. Interessant bei diesem Spiel ist auch wieder, daß es nahezu komplett vom PC übernommen wurde. So stammen auch die Grafik und die Menüleisten nicht vom ST-Betriebssystem, sondern sind selbst programmiert, was sich leider auch in der Geschwindigkeit bemerkbar macht. Das ist, von Sierra bekannt, ein Manko, das man in Kauf nehmen muß. Das Spiel selbst ist, wie bereits gesagt, recht kompliziert und sollte nicht von einem Adventure-Anfänger gespielt werden.

Sound ist, wie üblich, nicht vorhanden, doch das stört auch nicht weiter. Die Grafik des Spiels bewegt sich im gehobenen Bereich und ist ganz nett anzusehen, könnte aber auch besser sein. Alles in allem ist The Colonel's Bequest ein Adventure, das überdurchschnittlich ist und jedem empfohlen werden kann, der gerne Sierra-Adventures spielt. King's Quest-Liebhaber werden sich freuen, endlich einen würdigen Nachfolger zu haben - auch wenn das Spiel nicht so überzeugen konnte, wie man es sich vielleicht erhoffte.

MP



International 3D-Tennis



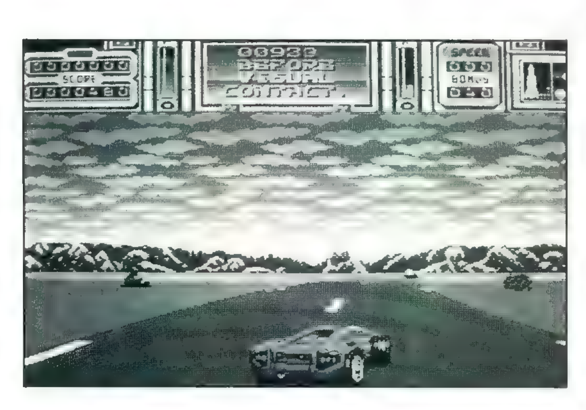
Tennisspiele gab es vor einigen Monaten en masse, da wundert es fast schon, daß Palace 'schon wieder' ein neues herausgebracht hat. International 3D-Tennis, so der Name des Probanten, kann allerdings mehr als andere Tennis-Simulationen, denn hier wird der Platz tatsächlich dreidimensional dargestellt. Außerdem existieren unterschiedliche Bodenbeläge, die natürlich auch jeweils das Verhalten des Spielballs beeinflussen. Unglaublich aber wahr: Das Spiel ist tatsächlich realistisch gelungen. Dazu tragen nicht zuletzt die digitalisierten Effekte bei. Wenn etwa der Sprecher „thirty - fourty" sagt, klingt das schon ziemlich echt. Auch die Anzeigetafel, auf der den Zuschauern immer der Punktestand bekanntgegeben wird, ist digitalisiert und trägt nicht unwesentlich zur Effektivität des Spiels bei. Wenn Sie den Blickwinkel auf das Spielfeld nicht mögen, können Sie ihn zu Anfang des Spiels auch mit den Funktionstasten verstellen. Doch eines bleibt dabei immer gleich: Ob links, ob rechts, es bleibt ein Tennis-Match.

Damit ist allerdings auch schon alles Interessante, womit das Spiel aufwarten kann, aufgezählt, denn der Rest des Games sieht weder gut aus noch macht er richtig Spaß. Die Vektorfiguren sind lahm wie Schildkröten und schleichen so langweilig über den Tennisplatz, als wären sie vor 5 Minuten erst aufgestanden. Der Ball fliegt über das Netz wie eine flügellahme Hausmücke, die eine Fliegenpatsche abbekommen hat. Kommt der Ball zur Figur, kann er geschlagen werden, sobald letztere aufblinkt. Nie zuvor habe ich blinkende Menschen gesehen. Vielleicht ist das die neue Mode? Leider geht der Ball unkontrollierbar ins Aus, wenn der Computer meint, nun sei die Zeit dazu gekommen. Interessanterweise läßt sich der Ball auch dann noch zum Gegner zurückbefördern, wenn er sich einen Meter hinter der Vektorfigur befindet. Hier scheint schlampig programmiert worden zu sein - eine so lange Rückhand kann es gar nicht geben. Boris Becker würde mit einem solchen Arm aussehen wie ein Monster. International 3D-Tennis ist wieder eines von den Spielen der Machart „Staubige Ecke, wo bist Du?". Nach anfänglichen 10 Minuten Spaß legt man die Diskette frustriert in die nächste Ecke oder formatiert sofort. Tip: Leerdisketten gibt's preiswerter.

MP



Fire and Forget II



„Fire & Forget II" spielt in ferner Zukunft. Die Hintergrundgeschichte berichtet von einem Überfall auf die Welthauptstadt Megalopolis, in der führende Staatsmänner gerade einen Friedenskongreß abhalten. Der Zufall will es so, daß sich der Spieler in Begleitung seines PS-starken Vehikels Thunder Master II in der Gegend aufhält. Der knallrote Rennwagen ist serienmäßig mit Bordkanone und Raketenwerfer ausgerüstet. Werden während der Fahrt genügend rote Kerosinkanister aufgesammelt, kann die Karre durch Druck auf die rechte Maustaste von der Fahrbahn abheben und den Gegnern in der Luft ihre Wendigkeit vorführen. Dies beeindruckt die Widersacher allerdings kaum: Die Armee der Terroristen umfaßt 50 unterschiedliche Vehikel, vom schwach gepanzerten Jeep am Ende des Konvois bis zum festungsartigen Führungspanzer an der Spitze. Fünf dieser rollenden Festungen müssen in den insgesamt 50 Levels besiegt werden, bevor Politiker und Wissenschaftler wieder ungestört am Weltfrieden basteln.

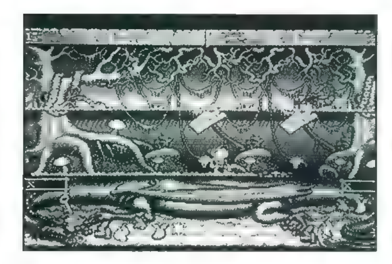
Zur Auflockerung sind Bonusrunden in den Spielverlauf eingeflochten. „Fire and Forget II" soll die fast perfekte Umsetzung einer Arcade-Maschine sein. Daß allerdings allzu viele Geldstücke in das Gerät gewandert sind, darf guten Gewissens bezweifelt werden. Technisch bewegt sich das Game auf dem Niveau von 1988. Die perspektivische Grafik ist ebenso schnell wie schlicht. Den Straßenrand säumen bestenfalls winzige Objekte, die viel zu schnell vergrößert werden. Da denkt man doch zu gerne an die C64-Zeiten zurück. Ein Gefühl für die Geschwindigkeit kommt nur durch die abwechselnd rot und weiß flimmernden Straßenbegrenzungen auf. Verirrt sich durch Zufall doch einmal ein größeres Objekt auf den Bildschirm, schaltet das Programm um zwei Gänge zurück. Dadurch wird das Geschehen nicht nur verlangsamt, sondern auch völlig unüberschaubar. Die größeren Gefährte werfen mit Geschossen um sich, die in der Hitze des Gefechts kaum als solche auszumachen sind.

Zudem hält sich die Abwechslung in arg engen Grenzen: hier Treibstoff, dort ein bißchen Munition einsammeln - und das war's dann auch schon. Wie der Titel schon eingibt: Dieses Spiel sollte man schnellstens verbrennen und dann vergessen!

CBO



Spell Bound



Meister Poinat des Alchimisten Sortie wird gekidnappt und in die Hölle verschleppt. Aufgabe: Den Lehrmeister finden und befreien. Der Weg des Zauberazubis führt durch horizontal-scrollende Landschaften, die durch ein verwirrendes Plattformgerüst miteinander verbunden sind. Anfangs sieht das Spiel vielversprechend aus. Doch das Scrolling setzt erst ein, wenn man den rechten Rand des Screens erreicht hat, und läßt einen oft in die winzigen Gegner hineinrennen. Schlimmer Flop!

CBO

3

Grafik

Sound

Motivation

Gesamtwertung

St Dragon



Das Spielprinzip ist allseits bekannt: Sie fliegen von links nach rechts und schießen dabei möglichst viele Gegner ab, die Ihnen natürlich in Strömen entgegenkommen. Weil es solcherlei Ballerspiele schon sehr häufig gibt, sind die Maßstäbe recht hoch gesetzt - und genau deshalb gibt's auch eine schlechtere Note, denn man kann wesentlich bessere Spiele als Saint Dragon kaufen. Die Grafik ist ganz nett, das Spiel selbst aber zu schwierig. Ein Spiel, das sich im oberen Durchschnitt bewegt - Anschauen lohnt sich also auf jeden Fall.

MP

5

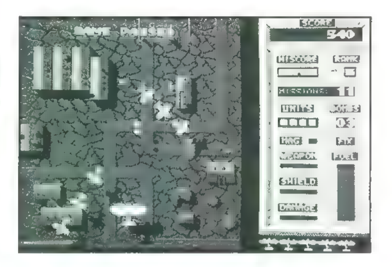
Grafik

Sound

Motivation

Gesamtwertung

Air Supply



Und wieder ein Ballerspiel in der Luft, bei dem man von unten nach oben fliegt. Die Besonderheiten: Stößt man an den Rand, wird das Flugzeug schneller. Außerdem müssen Sie Bodeneinheiten, die um Hilfe schreien, mit Versorgungspaketen ausstatten. Technisch ist das Spiel ganz nett gelungen, auch der Sound kann sich hören lassen. Wer Ballerspiele mag, ist mit Air Supply gut bedient. Wer nur ab und zu ballert, kann es getrost sein lassen, es gibt bessere Vertreter dieses Genres.

MP





Größere Bildschirmdiagonale

Beim SM 124 kann sehr einfach der Bildschirminhalt vergrößert werden. Der Rand wird dadurch entsprechend kleiner. Der Vorteil des Umbaus ist, daß der Bildschirminhalt deutlicher zu erkennen und auch aus größerer Entfernung sichtbar ist. Als Vorbereitung des „Umbaus" ist es ratsam, das Bildschirmfenster mit wasserfestem Filzstift zu markieren. Dann müssen die sechs Schrauben an der Monitorrückseite gelöst werden (Monitor vorher ausstecken). Der hintere Gehäuseteil kann dann vorsichtig herausgezogen werden, wobei es nötig ist, das Kabel zum Lautsprecher auf der Hauptplatine auszustecken. Suchen Sie danach auf der Hauptplatine den Trimmer VR601 (v size) und die Spule L 703 (h size). Drehen Sie solange, bis das Monitorbild jeweils ca. 1 cm über die Markierungen herausragt. Dabei sollten keine spannungsführenden Teile berührt werden, da im Monitor **lebensgefährliche** Spannungen vorhanden sind. Aus diesem Grund sollten auch gut isolierte Schraubenzieher verwendet werden. Nach dem Zusammenbau und Entfernung der Filzstiftmarkierungen haben Sie ein wesentlich besseres und größeres Monitorbild. Falls Sie Hardware haben, welche Ihr Monitor beeinflußt, können Probleme auftreten (ausprobieren).

Thomas Mokler, 7100 Heilbronn

Kontrastreicher Grafikdruck mit 1st_Wordplus

Sicher haben sich schon etliche AnwenderInnen von Wordplus geärgert, daß die Bildwiedergabe auf dem Drucker meist nur sehr unzureichend mit dem Bildschirmbild übereinstimmt. Neben Verzerrungen und Größendifferenzen tritt bei 24-Nadeldruckern auch eine sehr kontrastarme Druckdarstellung als Mangel in Erscheinung. Letzteres ist gegeben, wenn man, dem Handbuch zu Wordplus folgend, seinen 24-Nadler im Epson-Grafikmodus 0 (8 Nadeln, 60 dpi) betreibt. Da die Nadeln eines 24-Nadeldruckers nur etwa 1/120" Durchmesser haben, kann ein zu druckendes Pixel, das auf dem Papier etwa 1/60" x 1/60" groß ist, nicht voll geschwärzt werden. Dieser Umstand wirkt sich nur horizontal aus, vertikal werden immer drei benachbarte Nadeln pro Pixelhöhe benutzt. Man kann den Mangel aber sehr einfach beseitigen. Der Druckertreiber für Wordplus bietet nämlich eine Option, die es erlaubt, jedes Pixel zweimal nebeneinander drucken zu lassen. Das hat zwar auch zur Folge, daß sich die Bildbreite verdoppelt, läßt sich aber durch eine Verdoppelung der Grafikauflösung kompensieren. Wie das geht? Ganz einfach, der zweite Wert der sogenannten Druckergrundparameter muß den Wert 2 enthalten! Die Druckergrundparameter sind die ersten sechs Zahlen (HEX), die durch Kommata getrennt in einer Zeile der HEX-Datei stehen. Nun muß nur noch die Grafikauflösung verdoppelt werden. Dazu wählt man in Zeile 26 der Steuercodetabelle, diese folgt in der HEX-Datei unmittelbar auf die Druckergrundparameter, anstelle des Grafikmodus 0 den Modus 1 (8 Nadeln, 120 dpi). Nach der Umwandlung der HEX- in eine CFG-Datei hat man einen Druckertreiber, der zwar um 30% vergrößerte, aber formtreue und kontrastreiche Bilder auf dem 24-Nadeldrucker im Epson-Modus liefert.

Georg Schober, 8520 Buckenhof

Haben auch Sie einen *Quick-Tip*?

Standen Sie auch einmal vor einem kleinen, aber schier unlösbaren Problem? Dann, durch Zufall, bekamen Sie einen Tip - und schon war es gelöst.

Ähnlich haben wir uns diese neue Rubrik in der ST Computer vorgestellt. Aufgerufen sind auch Sie, liebe Leser(innen)! Geben Sie Ihre Erfahrungen weiter, egal, ob es um Anwendungen, Programmieren o.ä. geht.

Wir sammeln Ihre (und unsere) Tips und stellen Sie ggf. in den Quick-Tips vor.

Einsendungen an: MAXON Computer
ST Computer Redaktion
Stichwort: Quick-Tip
Industriestr. 26
D-6236 Eschborn

Umrüstung von 2 auf 6 Chips

Nach dem Umrüsten eines Mega 4 vom TOS-ROM in 2 Chips auf eines mit 6 Chips traten ständig nicht erklärbare Abstürze auf, eine Erscheinung, die sich bei der Benutzung des Spectre 128 noch verstärkte. Nachfragen bei einigen Bastlern führten schließlich zum Erfolg.

Durch Austausch von 2 TTL-Bausteinen ließen sich die Probleme dauerhaft beseitigen. Dabei handelt es sich um zwei Chips mit der Typenbezeichnung 74LS373 (es gibt nur 2 im ST). Diese müssen getauscht werden gegen zwei Chips mit der Bezeichnung 74ALS373. Diese Hochleistungs-Chips sind aber nicht überall zu bekommen, man kann es deshalb ersatzweise auch mit 2 Chips des Typs 74HC373 probieren. Oft sind diese schon leistungsfähig und schnell genug.

Die Probleme traten durch eine zu hohe Belastung des Busses auf. Die neuen Treiberbausteine sind leistungsfähiger als die alten und haben genug Reserven, um die zusätzliche Belastung durch die neuen ROM-Bausteine zu kompensieren. Tip: Beim Umrüsten die neuen Chips sockeln.

TW

AUTO-Ordner-Programme geordnet

Alle Programme im AUTO-Ordner werden nach dem Rechnerstart zuerst und automatisch ausgeführt. Dabei spielt die Reihenfolge oft eine wichtige Rolle. Um nun sicherzustellen, daß eine ganz bestimmte Reihenfolge auch tatsächlich eingehalten wird, sollte man folgendermaßen vorgehen: die Programmnamen so ändern, daß die gewünschte Reihenfolge durch vorangestellte Ziffern klar erkennbar ist, also z.B. „1QMOUSE.PRG“, „2FSELECT.PRG“, „3IMMUN.PRG“ usw. Nun werden die Programme in dieser Reihenfolge einzeln (!) in den AUTO-Ordner kopiert, damit die physikalische Reihenfolge mit der angezeigten übereinstimmt. Dadurch kann künftig beim Kopieren nichts mehr durcheinandergeraten. Das gilt natürlich nur, wenn man nach „Namen“ oder „Art“ (und nicht nach „Datum“ oder „Länge“) sortiert.

Rudolf Tiemann

Neuer Papierkorb

Das MAC-Desktop ist ausgereifter als das des ST. Das ist amtlich. Ab TOS 1.4 kann man mittels der CONTROL-Taste auch Files verschieben. Ich kam auf die Idee, mir, wie beim MAC, einen Mülleimer anzulegen, der die Daten behält. Ich legte mir auf meiner HD einen Ordner „MÜLL“ an und verschob darauf lustig meine zu löschenden Files. Aber das endgültige Löschen war doch etwas umständlich. Daher habe ich ein AUTO-Ordner-Programm geschrieben, das die Aufgabe für mich erledigt. So werden alle Dateien, die sich noch im „Mülleimer“ befinden, beim Einschalten oder nach einem Reset gelöscht. Beim TT oder beispielsweise mit Neodesk kann man den Ordner auch auf das Desktop legen und ihm sogar das Mülleimer-Symbol zuweisen!

Harald Wörndl-Aichriedler

```
1: PRINT „Lösche Papiereimer"
2: '
3: CHDRIVE 5              ! Laufwerk E
4: CHDIR „E:\MÜLL\"       ! Mull-Ordner
5: FILES „*.*"            ! File-Anzeige
6: WHILE EXIST(„*.*")     ! Loschschleife
7:   KILL „*.*"           ! Eliminieren
8: WEND
```

Listing: Wir löschen munter drauflos.

Abschalten des Resets

Mit einer kleinen Zeile in GFA-BASIC können Sie verhindern, daß beim Wechsel des Monitors (also dem Abtrennen des Monitorsteckers von der Buchse) ein Reset durchgeführt wird: SLPOKE&H46E, LPEEK (&H 408). Auf Adresse &H46E liegt der Vektor, über den gegebenenfalls gesprungen wird. Auf &H408 liegt ein Vektor auf ein RTN, der eigentlich zur Beendigung eines Programms dient. Wenn man diese Zeile compiliert und in den Auto-Ordner legt, gibt's keinen Reset mehr.

Doch Achtung: Bei einem richtigen Monitorwechsel kann der Shifter zerstört werden - vorher also wieder umprogrammieren!

Harald Wörndl-Aichriedler

USERDEF-Objekte leicht gemacht

Das AES bietet für selbstdefinierte Objekte den Typ USERDEF (ob_type=24) an. Für ein solches Objekt muß das eigene Programm die ganze Darstellung selbst übernehmen. Man kann sich dabei aber etwas Arbeit sparen. Wenn die Zeichenfunktion einen Wert ungleich 0 zurückgibt, wird dieser als ob_state interpretiert und das USERDEF-Objekt nachträglich verändert. Ist der Rückgabewert zum Beispiel 1 (SELECTED), wird es vom AES invertiert. Ist er 8 (DISABLED), wird das ganze Objekt mit einer Lighten-Maske „undiert“ und so weiter. Man muß also nicht selbst mühsam Routinen zum Invertieren, Umrahmen oder Schattenzeichnen entwickeln, sondern kann diese Knochenarbeit getrost dem AES überlassen.

Oliver Scheel

Weg mit störenden Dateien

Haben Sie sich auch schon einmal geärgert, daß durch die vielen Zusatzdateien zu Programmen die Verzeichnisse immer um ein Vielfaches länger werden, als sie eigentlich sein müßten? Ein paar Tricks schaffen hier Abhilfe. Bei normalen Files kann man im Attribut-Byte Bit 1 oder 2 (HIDDEN FILE bzw. HIDDEN SYSTEM FILE) setzen. *Fopen()* kann solche Dateien ohne weiteres öffnen. Voraussetzung ist aber, daß das Programm nicht zuerst mit *Fsfirst()* prüft, ob ein bestimmtes File vorhanden ist. Auch für die RSC-Dateien gibt es eine Möglichkeit zum Verstecken. Das AES sucht bei einem *rsrc*_load() mit *shel_find()* nach den gewünschten Dateien. Und diese Funktion verwendet erstaunlicherweise als Attributmaske %00010101, das HIDDEN SYSTEM-Bit ist also gesetzt! Folglich werden RSC-Dateien gefunden, die mit diesem Attribut versteckt wurden. Um seine Directories aufzuräumen, muß man also mit einem Programm, das das Ändern der File-Attribute erlaubt, alle störenden Dateien als HIDDEN SYSTEM kennzeichnen. Das Desktop zeigt sie dann nicht mehr an. Meistens werden die versteckten Dateien gefunden, im Zweifelsfall hilft ausprobieren.

Oliver Scheel

BAK-Dateien löschen

Der GFA-BASIC-Interpreter erzeugt leider (?) permanent BAK-Files. Diese zu löschen, ist keine nette Arbeit. Schreibt man sich in die DESKTOP.INF-Datei in etwa folgende Zeile:

```
#F 03 04  C:\CLR_BAK.PRG@
*.BAK@
```

dann werden bei einem Doppelklick auf die eine BAK-Datei alle BAK-Dateien gelöscht, die sich im Ordner befinden, sofern man als CLR_BAK.PRG Listing 4 installiert hat.

Harald Wörndl-Aichriedler

```
1: PRINT „Lösche BAK-Files"
2: PRINT
3: FILES „*.BAK"          ! Alle BAK-Files anzeigen
4: '
5: WHILE EXIST(„*.BAK") ! Löschschleife
6:   KILL „*.BAK"         ! Eliminieren
7: WEND
```

Listing: So werden die lästigen BAK-Dateien eliminiert.

Link-Virenschutz in GFA-BASIC

Der hier vorgestellte Link-Virenschutz funktioniert ganz einfach: Da ein Link-Virus die Programmlänge verändert, prüft das Programm einfach nach dem Start die eigene Länge und gibt ggf. Alarm. Das EXIST ist drin, damit kein Fehler -33 (Datei nicht gefunden) auftaucht, falls das Programm aus einer Shell gestartet wurde (dann funktioniert der Schutz natürlich nicht). Da man die Programmlänge vor dem Compilieren nicht kennt, gibt man zuerst eine ungefähre Zahl ein, bei der nur die Stellenzahl stimmen muß, compiliert, stellt im GFA-File dann die tatsächliche Länge ein und compiliert einfach nochmal. Den Programmnamen dürfen Sie nachträglich nicht ändern!

Rudolf Tiemann

```
 1: PROCEDURE test_linkvirus
 2:   LOCAL pfad$,prog$,al$,d.l%,antw&
 3:   pfad$=CHR$(GEMDOS(25)+65)+":"+DIR$(0)+"\"
 4:   ' der Pfad, von dem aus gestartet wurde,
 5:   ' wenn NICHT aus einer Shell gestartet
 6:   prog$=pfad$+"XYZ.PRG"
 7:   ' für „XYZ" muß der jeweilige Programmname
 8:   ' eingesetzt werden
 9:   IF EXIST(prog$)     !vermeidet ERR -33
10:     OPEN „I",#1,prog$
11:     d.l%=LOF(#1)
12:     CLOSE #1
13:     IF d.l%<>12345    !hier Länge einsetzen
14:       ' beim ersten Compilieren die richtige
15:       ' Stellenzahl, beim zweiten Compilieren
16:       ' die tatsächliche Länge einsetzen
17:       al$=" | Virusbefall | XYZ.PRG löschen|"
18:       ' hier wieder „XYZ" ersetzen
19:       ALERT 3,al$," weiter |Abbruch",antw&
20:       ' wer 'weiter' wählt, ist selbst schuld
21:       IF antw&=2
22:         END
23:       ENDIF
24:     ENDIF
25:   ENDIF
26: RETURN
```

Listing: Einfach, aber gut : ein Link-Virenschutz in GFA-BASIC.

Wer sind meine Eltern?

Manchmal ist es für ein Programm nützlich zu wissen, wie es gestartet wurde, ob durch den AUTO-Ordner, das Desktop oder ein weiteres Shell-Programm. Dazu sieht man sich die parent-Zeiger in der Basepage an. Dieser Zeiger verweist immer auf die Basepage des Prozesses, der das Programm gestartet hat. Nur der sogenannte Urprozeß, der bei der Initialisierung des GEMDOS angelegt wird, hat keine Eltern (wen auch?). Bei ihm ist der parent-Zeiger 0. Dieser Urprozeß ruft als erstes den Autostart-Prozeß auf, der alle im AUTO-Ordner enthaltenen Programme ausführt. Dieser Autostarter wird aber nie beim GEMDOS abgemeldet, so daß das anscheinend korrekt gestartete Desktop (oder COMMAND.PRG) sein Kind wird. Und so kann man aus der „Entfernung" des eigenen Programms zum Urprozeß Rückschlüsse auf seine eigene Position in dieser Hierarchie ziehen. Man zählt einfach, wie oft der parent-Zeiger ungleich 0 ist. Ist dies zweimal der Fall, befindet man sich in einem Autostartprogramm, bei dreimal sind weitere Shells zwischengeschaltet, die ihrerseits vom Desktop gestartet wurden. Das folgende Assembler-Programm liefert in D0 die Anzahl der Ebenen wie oben beschrieben. Es geht davon aus, daß *pbase* einen Zeiger auf die eigene Basepage enthält.

Oliver Scheel

```
 1: ; Offset des parent-Zeigers in
      der Basepage:
 2: parent      equ 36
 3:
 4: whatshell:  movea.l   pbase,a0
 5:             moveq     #0,d0
 6:
 7: loop:       tst.l     parent(a0)
 8:             beq       ende
 9:
10:             movea.l   parent(a0),
                          a0
11:             addq      #1,d0
12:             bra       loop
13:
14: ende:       rts
```

Listing: So wird die Anzahl der Ebenen bestimmt.


DISKETTEN-SERVICE
Absender bitte deutlich schreiben
Vorname / Name
Straße / Nr.
PLZ / Ort
Telefon
Postkarte
Bitte freimachen
MAXON Computer
Schwalbacher Straße 52
D-6236 Eschborn
ST COMPUTER
Kleinanzeigen
ST COMPUTER
Kleinanzeigen
Bitte freimachen
Absender
Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt 13
ST COMPUTER
ABO
ST COMPUTER
ABO
Bitte freimachen
Postkarte
Absender = Rechnungsadresse
Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt 13



Bitte senden Sie mir folgende PD-Disketten / Sonderdisks

Datum / Unterschrift

Zahlung erfolgt:
Vorauskasse (per Scheck) per Nachnahme

- Pro PD-Diskette entfällt ein Unkostenbeitrag von DM 10.-
- Preis der Sonderdisks je nach Angabe
- Bei Nachnahme zuzüglich DM 4,- Nachnahmegebühr
- Versandkosten (Porto und Verpackung) DM 5,- (Ausland DM 10,-)
- Auslandsbestellungen nur gegen Vorauskasse möglich

Ab 5 Disketten versandkostenfrei

SONDERDISK

Bitte beachten Sie, daß Sonderdisks ein Copyright besitzen und nicht weiterkopiert werden dürfen

DISKETTEN-SERVICE

Kleinanzeigen-Auftrag

Bitte veröffentlichen Sie für mich folgende Kleinanzeige in der angekreuzten Rubrik

Biete an	Hardware / Software	Ich suche	Hardware / Software	Tausch / Kontakte	Verschiedenes

30 Buchstaben je Standardzeile – incl. Satzzeichen und Wortzwischenräume
Groß- und Kleinbuchstaben verwenden, fettgedruckte Wörter unterstreichen.

Bearbeitung **nur gegen Vorausscheck** über den entsprechenden Betrag (keine Überweisung)

privat = kostenl.
gewerblich = DM 12,- je Zeile
Chiffregebühr = DM 10,-

Scheck über DM ______ ist beigefügt

Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze

Datum / Unterschrift



Kleinanzeigen



Abonnement

Kd.-Nr.:

Geschenk- bzw. Lieferadresse

Vorname

Name

Straße / Postfach

Land / PLZ / Ort

Bequem und bargeldlos durch Bankeinzug

Ein Verrechnungsscheck über DM ______ liegt bei.

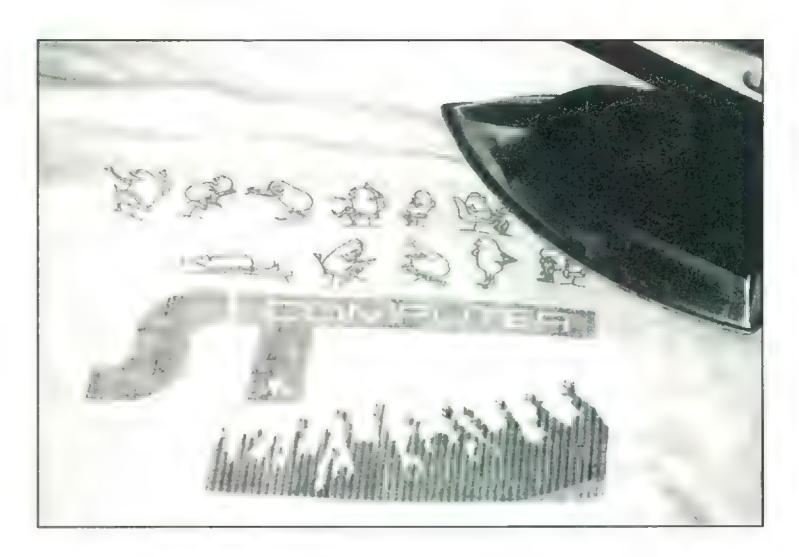
Vorauskasse per Zahlung auf unser Postscheck-Konto Ffm., BLZ 500 100 60, Kto.-Nr. ...

Diese Vereinbarung kann ich innerhalb von 8 Tagen beim Heim-Verlag, Heidelberger Landstr. 194, 6100 Darmstadt-Eberstadt widerrufen. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs. Ich bestätige die Kenntnisnahme des Widerrufsrechts durch meine 2. Unterschrift.



ABO

Diese Vereinbarung kann ich innerhalb von 8 Tagen beim Heim-Verlag, Heidelberger Landstr. 194, 6100 Darmstadt-Eberstadt widerrufen. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs. Ich bestätige die Kenntnisnahme des Widerrufsrechts durch meine 2. Unterschrift.



Magic Print

Das Medium ist die Botschaft

T-Shirts, im Sinne ihres Erfinders eigentlich Ausdruck (amerikanischer) Unterhemd-Kultur, avancierten schon in den 60ern zum Träger von (Werbe)-Botschaften aller Art. Seit den 80er Jahren sind T-Shirts als persönliches Medium in unserer Gesellschaft voll akzeptiert und der Aufdruck eines T-Shirts muß heute so individuell wie die Persönlichkeit seines Trägers sein. Problematisch sind allein die Beschaffung und letzendlich der Preis für soviel modische Idividualität.

Der schlaue Computerbenutzer mit Drukker ist dem Normalverbraucher hier um eine Nasenlänge voraus. Er entwirft sich die Motive mit einem Grafikprogramm einfach selbst, druckt sie aus und bügelt sie auf weiße T-Shirts.

Heißes Eisen

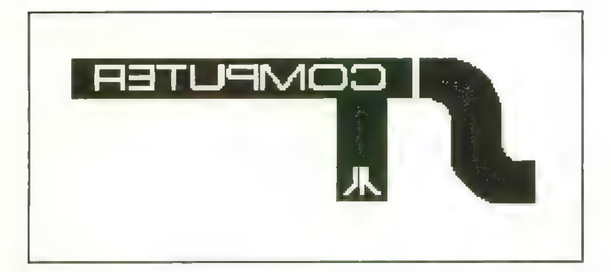
Die Firma Unicomp bietet Farbbänder an, mit denen man exzellente Ausdrucke erstellen kann, die auf die unterschiedlichsten Materialien aufgebracht werden können. Einfachstes Beispiel ist in diesem Fall ein T-Shirt, z.B. aus Baumwolle oder Polyester (oder einer entsprechenden Mischung). Was macht also der geneigte Individualist, wenn er eine nette Grafik auf seinem Bildschirm entdeckt? Ganz einfach: Farbband einsetzen, ausdrucken, aufbügeln. Mehr braucht es wirklich nicht, um seinen Monitor auf die Kleidung zu bannen. Was dazu benötigt wird, ist ebenfalls nicht viel. Ein Bügeleisen (Stufe 3, Baumwolle), Papier und natürlich die passenden Farbbänder - und schon kann's losgehen.

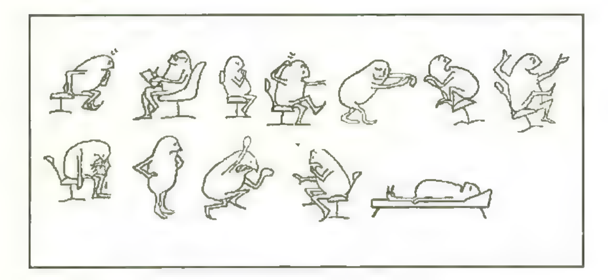
Nun wäre es recht langweilig, nur mit Computerbildern durch die Stadt zu rennen. Natürlich ist es auf diese Art und Weise sogar möglich, seine bevorzugten Plattencovers einzuscannen, und schon hat man ein Kleidungsstück, um das man bei jedem Konzert beneidet wird. Geschäftsnaturen nehmen für solche Sammlerstükke mehr als 100 Mark - wer würde da nicht gerne mitverdienen?

Durstlöscher

Wem die Prozedur zu langweilig wird, der nehme sich eine Tasse und den Fixierlack von Unicomp. Auch hier geht's wieder ganz einfach: Lack auf die Tasse streichen, Ausdruck auflegen und festkleben, Tasse in den Backofen. Nach dem „Bakken" den Ausdruck abnehmen, erneut Fixierlack aufstreichen und nochmals „bakken". Schon ist die individuelle Tasse fertig. Auf diese Art und Weise ist es sogar problemlos möglich, sich ein komplettes Geschirr mit individuellem Aufdruck zu erstellen, um das Sie jeder beneiden wird. Wer hat schließlich schon Geschirr mit „Commodore"-Aufdruck zu Hause? Das Commodore-Logo macht sich auf Tassen, Tellern, Gläsern und sogar auf Besteck wirklich gut!

Wie funktioniert denn nun die ganze Geschichte? Ganz einfach: Im Farbband ist ein spezieller Stoff enthalten. Dieser wird natürlich auch auf den Ausdruck übertragen. Wird letzterer aufgebügelt, geht der Ausdruck direkt über in den gasförmigen Zustand und läßt sich nicht mehr ablösen. Wer's nicht glaubt, wasche sein T-Shirt 30mal und wird feststellen, daß die Farben noch genau so aussehen wie nach dem ersten Bügeln (ich habe selbst gezweifelt und es ausprobiert: funktioniert tatsächlich). Sofern Sie Ihr Geschirr durch Ausdrucke modifizieren wollen, kann ich Sie auch hier beruhigen: Durch den Fixierlack werden alle Motive kratz- und spülmaschinenfest. Durch die Struktur, die z.B. Baumwolle oder Polyester aufweist, werden Ungenauigkeiten beim Ausdruck ausgemerzt. Man kann also auch problemlos zackelige Motive ausdrucken. Beim Aufbügeln verteilen sich alle Ecken und Kanten - was natürlich nicht heißen soll, daß der gesamte Ausdruck schwammig wird. Einen Beispielbügeldruck sehen Sie am besten auf unserem Foto.







Bilder 1-3: Die verwendeten Vorlagen für unser Probe-Shirt

Coca Cola

Sicher kennen Sie auch die Kult-Spiegel (z.B. von Coca Cola), die man sich so gerne in Party-Kellern an die Wand hängt? Wer für diese Geräte jetzt noch Geld ausgibt, dem kann nicht mehr geholfen werden - es geht viel einfacher! Coca-Cola- oder anderes Signet einscannen, ausdrucken, auf einen leeren Spiegel bringen, fertig ist der fesche Spiegel. Doch nicht nur das. Sie können sogar Motive entwerfen, die im freien Handel nicht erhältlich sind. Vielleicht sehen Sie morgens lieber einen anderen Menschen im Spiegel als sich selbst? Kein Problem, der/die Freund/in läßt sich natürlich ebenso gut auf den Spiegel aufdrucken.

Die Anwendungsmöglichkeiten des Farbbandes sind quasi unbegrenzt. Sei es Kleidung, Spiegel, Geschirr, Teppiche, Fenster, Sturmfeuerzeuge, alles kann jetzt bedruckt werden. Gerade zur Weihnachtszeit dürfte ein großer Bedarf an Magic Print-Farbbändern herrschen, denn viele tolle Geschenkideen lassen sich erst durch dieses Produkt verwirklichen. Vielleicht versenden Sie Ihre Weihnachtsgrüße ja auch mal auf ein T-Shirt aufgebügelt? Sicherlich eine tolle Idee! Die Farbbänder kosten nicht so viel, wie man jetzt denken sollte. Der Durchschnittspreis liegt bei etwa 20 Mark, einige sind preiswerter, einige teurer. Es gibt sie in den Farben Rot, Blau Gelb und Schwarz. Ab März sollen dann auch Vierfarbbänder für die gängigsten Farbdrucker erhältlich sein.

Wer's nicht glaubt, sollte sich selbst überzeugen. Magic Print bringt Ihre Ausdrucke auf die unglaublichsten Materialien, und das zu einem guten Preis. Vorbei ist die Zeit der teuren Werbematerialien. Übrigens: Die Unicomp-Farbbänder werden nach dem sogenannten ALQ-Verfahren geprüft. Das ist ein spezielles Prüfverfahren für Farbbänder. Wer Probleme mit seinen Bändern hat, kann sie kostenlos umtauschen.

MP

Bezugsadresse:

Unicomp GmbH
Feldmarkring 233
5860 Iserlohn
Tel. (02371) 41081



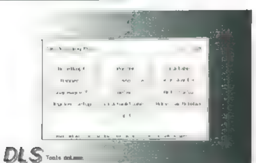
Der Portfolio macht Furore: Seit der Vorstellung des Interfaceprogramms FolioTalk – samt Verbindungskabel zum ATARI ST – in der Take 20 Anzeige, dockt der kleinste PC der Welt immer häufiger an seinen großen Bruder an. Damit die Verständigung zwischen DOS und TOS noch interessanter wird, haben wir eine Diskette mit Folio PD zusammengestellt. Jetzt hat der Kleine sogar was zum Spielen.

Die Spielepackung vom Vormonat – 2171 bis 2180 – kommt bei ST Anwendern sehr gut an. Ein Grund mehr, sich mit der Programmierung anspruchsvoller Unterhaltungssoftware zu beschäftigen. Pixie von Diskette 2185 ist ein geniales Utility, das den Einstieg in dieses Genre erheblich vereinfacht. Viel Erfolg damit.

Mit freundlichen Grüßen,

Die "23"

2181



The Printing Press 3.2 hat gegenüber der Version 3.1 wieder einige Verbesserungen erfahren: Die Benutzeroberfläche zeigt sich jetzt im neuen Design (s/w, MB, S).

2182



Visitenkarten können Sie jetzt in dem gleichnamigen Printing Press Modul gestalten und ausdrucken. Die Bedienung ist verblüffend einfach und erlaubt dem Anwender, mit verschiedenen Designs zu experimentieren (s/w)

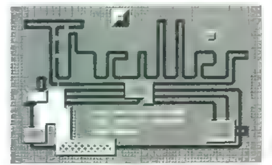
Formular II von Knightsoft erstellt Formulare oder fügt Texte in bereits vorhandene Formulavordrucke ein. Bei der horizontalen und vertikalen Positionierung arbeitet Formular mit einer Genauigkeit von 1/12 Zoll (s/w)

Minitext 2.79 hat in den letzten Monaten viele neue Funktionen erhalten, die aus der "kleinen" Textverarbeitung ein immer professionelleres Werkzeug machen (s/w)

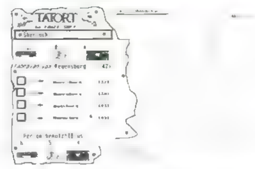
SPS 2.2k emuliert in der Industrie verwendete SPS-Geräte. Mit SPS erstellte Anweisungsliste können im Programm abgearbeitet und getestet werden (s/w)

2183

Thriller gehört zur Serie der anspruchsvollen und grafisch gut umgesetzten Denk- und Geschicklichkeitsspiele, die mit OXYD, Gulp und Future Garden schon viele Freunde gefunden hat. In Thriller wird die Spielkugel auf fest vorgegebenen Rollbahnen bewegt, die unter anderem auch von aggressiven Killerkugeln befahren werden Um nicht überrollt zu werden, muß der Spieler seine Kugel immer wieder in eine der Haltebuchten schieben und die Verfolger an sich vorbeiziehen lassen (s/w)



2184



Tatort schickt 2-4 Detektive auf Verbrecherjagd. Der Täter reist mit Bus, Bahn oder Flugzeug von einem Versteck zum nächsten. Gelegentlich gehen Meldungen ein, die seinen Aufenthaltsort und das soeben benutzte Verkehrsmittel enthalten. Wer jetzt den nächsten Spielzug des Gauners errät, kann mit dessen Verhaftung rechnen

PacBalls ist Martin Senftlebens witzige Neuauflage des klassischen Themas. Diamanten, Käsestückchen, Beam-Buttons, der eingebaute Level-Editor und viele weitere Gags geben dem alten Game neuen Inhalt (s/w)

Mondlander verlangt Feingefühl und räumliches Vorstellungsvermögen vom Spieler Eine Raumkapsel soll möglichst sanft auf dem Mond aufgesetzt werden (s/w)

2185

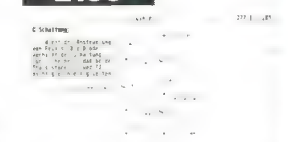


Fortress ist eine Bastion im Weltraum, von der aus die Erde gegen Space-Monster verteidigt wird Der Festungskommandant muß sich gegen immer neue Angriffswellen behaupten (f)

Seapower versetzt den Spieler in die Feuerleitzentrale eines Schlachtschiffes Anhand der Zielbesprechung muß das Feuer der eigenen Geschütze korrigiert werden, damit der nächste Schuß im Ziel liegt (f)

Pixie ist ein fantastisches Hilfmittel für Programmierer von Grafik-Abenteuern. Bildausschnitte gleicher Größe werden in einer gemeinsamen Bilddatei zusammengefaßt, Pixie erstellt eine dazu passende Laderoutine im LST-Format von GFA 3.0+. Wie's geht, zeigt Markus Kronenberg in seinem Beispieladventure **Odysee**

2186



GEM A4 Graf 2.11 erstellt Zeichnungen mit 1440 x 1984 Bildpunkten, die im gedruckten Zustand einer A4-Seite entsprechen. Fadenkreuz, eingeblendete Bildschirmkoordinaten und WYSIWYG-Prinzip machen GEM A4 Graf zu einem präzisen Zeichenwerkzeug. Im IMG-, PIC-, DOO-, PI3-, PAC-, SEQ- oder PCB-Format gespeichert, können die Bilder in vielen anderen Programmen weiterverarbeitet werden (s/w, MB)

Die **PCB-Library** enthält zahlreiche Platinenlayouts im 100mm x 160mm großen Europakartenformat Die Layouts können mit PCB-Edit von Diskette 2027 bearbeitet werden (s/w)

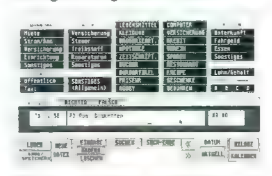
Das **PCB-Bohrprogramm** liest PCB-Dateien ein, ermittelt Bohrungen, Bohrdurchmesser und -formate, die als DRL-Datei im ASCII-Format abgespeichert und bei der Platinenherstellung verwendet werden (s/w)

Besonderheiten:

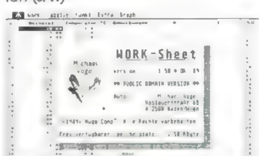
f	= Läuft nur in Farbe
s/w	= Läuft nur monochrom
E90	= Läuft auch mit Emula 5.1 von Diskette 2090.
e	= in englischer Sprache
J	= Joystick notwendig
MB	= 1MB RAM erforderlich
S	= Shareware
G	= GFA-Quellcode liegt bei
K	= Kontaktkarte eingebaut

2187

Folio PD ist eine ganze Sammlung nützlicher Programme für den kleinsten PC der Welt Die grafische Datenverwaltung FOLIBASE gehört dazu, außerdem einige nützliche Utilities, Demos, Updates und ein Spiel. Wichtig für ST-Anwender: Zur Datenübertragung vom Folio auf den ST stehen gleich zwei gute Programme bereit

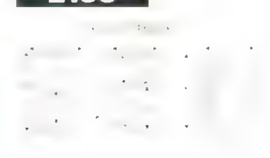


Bilanz verwaltet sämtliche Ausgaben für Wohnung, Fahrgeld, Auto, Reisen und sonstige Aktivitäten Jede Ausgabe kann einer von 35 Rubriken zugeordnet werden: Die Einzelposten geben dann Aufschluß über das Konsumverhalten (s/w)



Work ist ein kleines Kalkulationsprogramm mit 31 Zeilen und 5 Spalten. Damit lassen sich vielfältige Berechnungen auf einfache Weise durchführen und grafisch darstellen: Von der Haushalts- und Kfz-Kostenrechnung, bis hin zu wissenschaftlichen Anwendungen

2188



High End 3 unterstützt Bastler beim Eigenbau von Lautsprecherboxen. Das Programm führt Be-

rechnungen verschiedener Boxentypen durch, bis hin zur Erstellung von Materiallisten. High End wird seit Jahren regelmäßig erweitert und gepflegt (S, MB, s/w)

2189

Das **Benutzer Lexikon** von Bernd Fritz ist 66502 Bytes groß und enthält Fachbegriffe aus verschiedenen Geschäftsbereichen (Nur in Verbindung mit "Rechtschreibprofi" von Data-Becker verwendbar).

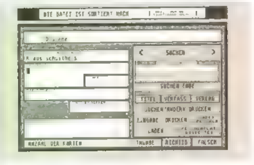
Proportional und **Proportional Breit** sind neue Fonts für Anwender der Text-Verarbeitung That's Write. Das Fontpaket umfaßt die benötigten Drucker- und Bildschirmfonts.

WPrint druckt Ist WordPlus Dokumente mit frei wählbaren Zeilenabständen aus. WPrint verwaltet den Seitenumbruch eines Dokuments und führt eine Feinpositionierung durch

Fußnote-Endnote ermöglicht die Ausgabe von WordPlus Dokumenten in zwei verschiedene Dateien: Texte werden in eine DOF-Datei geschrieben (Dokument ohne Fußnoten) und die Fußnoten in der FNO-Datei gesammelt.

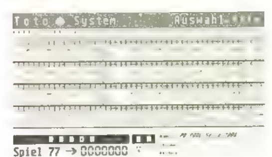
CfgInstall wandelt HEX-Dateien in CFG-Format und umgekehrt Damit ist es möglich, umfangreiche Änderungen an den Treibern vorzunehmen

2190



Archiv verwaltet Bücher, Zeitschriften und viele andere literarische Erzeugnisse Das Programm zeichnet sich durch professionelle Gestaltung, sinnvolle Funktionen und leicht verständlicher Bedienung aus: Wieder ein durchdachtes Softwareprodukt von Dipl.-Ing. Jürgen Pfeng (s/w, S)

Bausparen 2.0 berechnet den Verlauf eines Bausparvertrags bezogen auf Sparzahlungen unterschiedlicher Höhe. Wann ein Vertrag zuteilungsreif wird, läßt sich im Rahmen verschiedener Modelle und unterschiedlicher Vertragsbestimmungen von 7 Bausparkassen ermitteln. Diese Version überprüft sogar die Einhaltung wichtiger Vertragsmodalitäten (s/w).



Totokat 1.0 überläßt nichts dem Zufall: Das Programm verwaltet sämtliche Ziehungen der Auswahlwette 6 aus 45, wertet sie statistisch aus und stellt die Ergebnisse grafisch dar. Wenn Normal- und Systemscheine am Bildschirm ausgefüllt werden, ermittelt Totokat den vermutlichen Gewinn. Eine Gewinngarantie gibt das Programm zwar nicht, wer nicht gerne im dunklen Tippt ist damit aber dennoch gut beraten (s/w, S).

Kartei verwaltet Daten aller Art in Form eines Karteikasten. 1000 Kärtchen stehen dem Anwender zur Verfügung. Kartei ist ansprechend gestaltet, arbeitet sehr schnell und läßt sich leicht bedienen (s/w).

Profibase 0.24 hat seit der letzten Pool-Version (Disk 2010) zahlreiche Verbesserungen und Funktionserweiterungen erfahren. Jetzt wurde das Programmdesign überarbeitet und ein eigener Zeichensatz eingebaut. Wer eine professionelle Arbeitsumgebung zur Datenverwaltung benötigt, für den ist Profibase genau richtig. Das Programm wird von Elmar Zipp liebevoll gepflegt (s/w, S).



Jahresinhaltsverzeichnis

Als Leser Ihrer Zeitschrift seit der Ausgabe 1/86 habe ich ein kleines Problem mit Ihren Jahresinhaltsverzeichnissen. Da ich nur die Programmiersprache Pascal beherrsche, benutze ich seit einiger Zeit relativ häufig die Jahresinhaltsverzeichnisse der Jahrgänge 1986 bis 1989, um mir Anregungen und Informationen bei der Programmierung meiner Probleme zu holen. Das Handling wird dabei dadurch erschwert, daß in den Titeln der Listings häufig nicht vermerkt ist, in welcher Programmiersprache das betreffende Programm ausgeführt worden ist.

Manfred Köhler, W-3500 Kassel

Red.: Ihren Vorschlag nehmen wir gerne auf. Ab sofort werden Sie im Jahresinhaltsverzeichnis die Listings nur noch mit Angabe der verwendeten Programmiersprache finden.

*

Rechnender Drucker

Ich besitze einen Atari 1040 und als Drucker dazu einen Seikosha SL 80 IP. Zur Textverarbeitung benutze ich das Programm Signum!2. Beim Ausdrucken meiner Texte stellte ich fest, daß der Computer (oder der Drucker) fast eine Sekunde Zeit für das Berechnen einer vollgeschriebenen Zeile braucht, was ich von Druckern meiner Bekannten nicht kenne (NEC P6). Der eigentliche Druckvorgang geht relativ schnell vonstatten. Gibt es ein Programm, das meinen Drucker veranlaßt, die einzelnen Zeilen schneller zu berechnen?

Thomas Schmidt, W-2300 Kiel

Red.: Einige Drucker brauchen eine gewisse Zeit, um die gesendeten Daten „druckkopfgerecht" aufzubereiten - der eine mehr, der andere weniger. Diese Zeit können Sie leider nicht verkürzen.

*

STE-Sound?

Seit einiger Zeit bin ich stolzer Besitzer eines Atari STE mit 2 MB. Ich habe mich für den STE entschieden, weil ich über die PCM-Klangerzeugung des STE Sounds programmieren wollte. Leider konnte ich zu diesem Thema bis heute keine Literatur finden. Deshalb meine Bitte: Können Sie mir ein Buch oder einen Artikel nennen, in dem die Klangerzeugung des STE beschrieben ist?

Peter Engler, W-7015 Korntal

Red.: Uns ist kein Fachartikel bekannt, in dem das Problem geschildert wird. Grundsätzliche Informationen finden sich aber in der STE-Dokumentation von Atari. Für weitere Infos sollten Sie sich direkt mit Atari Deutschland in Verbindung setzen, zuständig dafür ist Herr Müller.

Schneide-Plotter

Können wir mit unserem Mega ST2 mit entsprechender Soft- bzw. Hardware einen Schneide-Plotter betreiben? Wenn ja, mit welcher?

Messebau Boldt&Liedecke, W-5100 Aachen

Red.: Ein Schneide-Plotter ist im Prinzip nichts anderes als ein normaler Plotter, der statt eines Schreibstifts ein Messer eingebaut hat. Allerdings gibt es auch spezielle Schneide-Plotter, die Sie z.B. über die Firma GmA aus Hamburg erhalten.

*

LCD-Monitore

Für meinen ST suche ich nach einem LCD-Bildschirm, wur-

Ein Wort in eigener Sache

In den Jahren, die unsere Zeitschrift existiert, haben wir immer wieder versucht durch die Beantwortung der bei uns eingehenden Briefe ein wenig Licht in das Dunkel zu bringen, das bei der Arbeit mit dem ATARI ST schon so manch einen aus der Fassung bringen konnte - eine Tatsache, die nicht nur Ihnen, verehrter Leser, sondern auch uns oft genug zu schaffen machte Nichtsdestotrotz haben wir uns bemüht, die Probleme zu lösen und diverse Leserbriefe zu veröffentlichen, da wir der Meinung waren, daß die jeweilige Thematik auch einen größeren Leserkreis interessieren könnte. Trotzdem gibt es immer wieder Briefe, die wir nicht beantworten können oder durfen. Damit Sie nicht allzusehr enttauscht zu sein brauchen oder keine Antwort erhalten, möchten wir Sie bitten, sich an folgende Spielregeln zu halten, die sich aus unserer Erfahrung ergeben haben. Fällt Ihr Brief nicht unter die folgenden Kriterien, hat er gute Chancen positiv beantwortet oder wenigstens als Hilferuf an unsere Leserschaft gedruckt zu werden

1. Leider gehen immer wieder Briefe mit dem Wunsch ein, ein Produkt für diesen oder jenen Anwendungsfall vorzuschlagen, verschiedene Produkte bezüglich der Vor- und Nachteile gegeneinander abzuwägen und zu bewerten. Es ist uns aus Wettbewerbsgründen nicht erlaubt ein bestimmtes Produkt zu favorisieren, selbst wenn wir das eine oder andere in der Redaktion uberzeugt einsetzen Wir können Sie in diesem Fall ausschließlich auf die von uns möglichst objektiven Tests und eventuell anstehende Fachmessen hinweisen. Bedenken Sie bitte, daß auch wir nicht jede Textverarbeitung, jedes Malprogramm und so weiter kennen und bestimmte Produkte dadurch in das Abseits drängen würden

2. Oft erreichen uns Briefe, die sich positiv oder auch negativ über bestimmte Händler, Softwarehäuser oder deren Produkte auslassen. Sicherlich interessieren uns solche Bemerkungen Bitte haben Sie aber Verständnis, daß wir weder Lob noch Tadel abdrucken durfen, da diese Aussagen meist subjektiv sind Anders sieht die Sache beispielsweise bei Gerichtsurteilen aus, die Sie, verehrte(r) Leser(in), erfochten haben

3. Aufgrund der Vielzahl an Briefen, die uns täglich erreichen, sind wir leider nicht in der Lage, Programmfehler anhand von Listings oder ähnlichem zu korrigieren. Dennoch sollte ein Problem möglichst detailliert beschrieben sein, denn Ferndiagnosen sind prinzipiell sehr schwer, jedoch mit genauerer Angabe der Symptome eventuell durchfuhrbar

4. Von Zeit zu Zeit erreichen uns Briefe mit der Bitte, die Adresse des Lesers zwecks allgemeiner Kontaktaufnahme zu veroffentlichen Würden wir dies in die Tat umsetzen, wurde sich der Umfang des anderen redaktionellen Teils beträchtlich verkleinern. Ausnahmen stellen Leser in fernen Ländern dar, für die eine Kontaktaufnahme im eigenen Land recht schwierig ist

Zum Schluß sollen ein paar Tips eventuell voreilig geschriebene Briefe verhindern

1. Wenn Sie ein Problem bezüglich einer bestimmten Problematik haben oder an einem bestimmten Produkt interessiert sind, finden Sie interessante Artikel darüber eventuell in vorhergehenden Ausgaben userer Zeitschrift Zur Auswahl eignet sich das Jahresinhaltsverzeichnis besonders gut, das immer am Jahresende in der ST Computer abgedruckt wird

2. Sollten die Probleme mit der Handhabung eines Produktes zu tun haben, wenden Sie sich zunächst an Ihren Händler und über diesen an den Distributor beziehungsweise an das Software-Haus. Die Wahrscheinlichkeit, daß Ihnen das Software-Haus weiterhelfen kann ist um ein Vielfaches höher als die, daß wir Ihnen helfen können.

3. Lesen Sie aufmerksam die Leserbrief-Seite Viele Fragen wiederholen sich immer wieder obwohl wir bestimmte Probleme schon mehrfach angesprochen haben

de aber bisher nicht fündig. Ich kann mich erinnern, daß in der ST-Computer mal einer vorgestellt wurde, es war mir aber trotz Sichtens umfangreicher ST-Berge nicht vergönnt, den Artikel wiederzufinden. Können Sie mir entsprechende Fabrikate nennen?

Christoph Pahl, W-8201 Riedering

Red.: Entsprechende Geräte erhalten Sie z.B. bei CrystalVision, Weberstraße 24, W-5300 Bonn, oder bei der invent ag, Wehntalerstraße 272, CH-8166 Niederweningen.

*

Scharfes S in Script

Ich besitze einen 1040 ST mit der c't-Billiglösung und einem TOS 1.4 mit dem Omti-Patch. Ferner habe ich mir Script gekauft und habe nun damit ein Problem, bei dem mir auch Application Systems nicht helfen kann. Ich komme nicht an das scharfe 's' heran. Da ich keinen Zeichensatzeditor mein eigen nenne, kann ich mir die Zeichensätze auch nicht umdefinieren. Für mich wäre daher wichtig zu erfahren, ob vielleicht einer Ihrer Leser auch so ein Problem (gelöst) hat, und/oder woran der Fehler liegen könnte.

Gerhard Schneider, W-8526 Bubenreuth

Red.: Ein solches Problem kann eigentlich nur eine Ursache haben: In Ihrem TOS ist nicht nur ein Omti-Patch, sondern zusätzlich auch noch einer für die Tastatur. Das scharfe 's', das normalerweise den ASCII-Wert 158 hat, wird bei Ihnen den Wert 225 erzeugen, der in Script meistens nicht definiert ist. Abhilfe: Entfernen Sie den Patch aus den ROMs oder benutzen Sie ein Programm, mit dem Sie die Tastatur umdefinieren können, etwa Harlekin oder andere.

*

PD-Modula und XBIOS

Bei der Benutzung des PD-Modula (PD 225) vermisse ich die angekündigte Implementation der XBIOS-Funktionen. Es befindet sich zwar ein unvollständiges Modul auf der Diskette, doch existiert auch ein ausführliches und vollständigeres Definitionsmodul in dem Verzeichnis DEF\GEMLIB. Aber wo befindet sich das zugehörige Implementationsmodul?

Holger Dormann, W-3052 Bad Nenndorf

Red.: Das Modul befindet sich auf einer Folgediskette (PD 334).

*

Mega-Tastatur

Zur Zeit baue ich meinen 1040 ST in ein selbst gefertigtes Gehäuse um, da durch zahlreiche Erweiterungen im Ursprungsgehäuse Platz- und Wärmeprobleme auftreten. Nach diesem Umbau kann ich leider nicht die ursprüngliche Tastatur benutzen. Läßt sich die Tastatur des Mega ST soft- oder hardwaremäßig auch an den 1040 anschließen?

Norbert Linde, W-7146 Tamm-Hohenstange

Red.: Die Mega-Tastatur läßt sich absolut problemlos an jeden ST anschließen. Beim 1040 verbinden Sie folgende Pins (jeweils Mega ST-Tastaturkabel/1040): 6/8, 5/8, 4/5, 3/6, 2/4, 1/4. Eine genaue Bauanleitung finden Sie in Ausgabe 10/90 auf Seite 162.

*

GFA-BASIC unter DOS?

Zur Zeit besuche ich eine Fachschule für NC-Anlagentechnik. Mein Computer (520 STM aufgerüstet mit PC-Speed) ist mir natürlich dabei eine große Hilfe. Aber das Problem ist, daß ich die Programme, die ich schreibe, auch gerne meinen Mitschülern, sie besitzen meist einen XT, zugänglich machen will, aber wie? Alle BASIC-Versionen, die ich bis jetzt für MS-DOS-Rechner gesehen haben, sind nicht kompatibel zu GFA-BASIC. Ich möchte die Sprache aber nicht wechseln. Ist Ihnen vielleicht eine Möglichkeit bekannt?

Klaus-Peter Mathes, W-5760 Arnsberg

Red.: Seit kurzer Zeit gibt es GFA-BASIC auch für IBM-kompatible Computer, bzw. das Programm ist von GFA Systemtechnik bereits angekündigt und wird auch schon beworben. Sollte es noch nicht fertiggestellt sein, können Sie ersatzweise auch Turbo-Basic benutzen, das zu etwa 80 Prozent kompatibel ist. Die wichtigsten Befehle stimmen überein. Turbo-Basic ist inzwischen auch unter dem Namen Power Basic bekannt, wird allerdings vom Hersteller, Borland, nicht mehr supportet. Als ST-Alternative bietet such auch



Helmut Kopka
LaTeX-Erweiterungsmöglichkeiten mit einer Einführung in METAFONT

Bonn 1990
Verlag Addison-Wesley
463 Seiten, DM 79,
ISBN 3-89319-287-5

Kopkas Buch schließt sich an die LaTeX-Einführung desselben Autors an und wendet sich an den geübten LaTeX-Anwender, dem die vorgegebenen Möglichkeiten nicht mehr ausreichen.

Im ersten Teil beschreibt Kopka einige Standardergänzungen, die das normale LaTeX-Paket erweitern. Wichtig für den deutschsprachigen Raum ist der GERMAN-Style von Hubert Partl, mit dem beispielsweise ff nach ff-f oder ck nach k-k getrennt wird. Kopka beschreibt ihn in seiner Anwendung und geht auch auf die interne Struktur des Style-Files ein.

Es schließen sich Beschreibungen der erweiterten tabular-Umgebung von Frank Mittelbach und der supertabular-Erweiterung von Gabriele Kruljac-Dronskowski an. Angesichts der Unmengen von LaTeX-Ergänzungen hat Kopka eine gelungene Auswahl getroffen, beschreibt aber abschließend noch sehr praxisorientiert den Netz-Zugang zu den Dateisammlungen der TUG- und DANTE-Anwendervereinigungen.

Die folgenden Kapitel beschäftigen sich mit der LaTeX-Einbindung der mathematischen Sonderzeichen des AMSTeX-Pakets und der Anwendung des SLITeX-Formats für die Erstellung von Overhead-Folien. Wichtig ist auch die Beschreibung des Make-Index-Programms, mit dem LaTeX-Register sehr komfortabel sortiert und gestaltet werden können.

Im vierten Kapitel findet sich ein ganz besonderer Leckerbissen, da hier das PiCTeX-Paket von Michael Wichura beschrieben wird. PiCTeX ermöglicht wirkliche Grafikanwendungen mit TeX - so können beispielsweise freie Kurven gezogen, Flächen mit Rastern unterlegt, Linien beliebig gestrichelt oder auch Diagramme mit einer sehr flexiblen Achsengestaltung versehen werden. PiCTeX ist - wie die oben genannten Makro-Sammlungen - frei erhältlich, leider war eine Dokumentation bisher nur in den USA über die TUG zu bestellen.

Kopka beschreibt das Paket anschaulich auf 74 Seiten, die mit vielen Beispielen und Illustrationen versehen sind. Das Kapitel wird abgeschlossen mit einem Index, der nicht nur auf eine Seitenzahl verweist, sondern sich durch eine kleine Erläuterung hervorragend zum Nachschlagen eignet. Eine ähnliche Registergestaltung hat sich schon bei den anderen TeX- und LaTeX-Büchern von Addison-Wesley bewährt.

Damit endet die Anwenderbeschreibung existierender Erweiterungen und der zweite Teil des Buchs führt den Leser zu dem Wissen, das man für eigene LaTeX-Erweiterungen benötigt. Es beginnt mit einer detaillierten Beschreibung der verschiedenen TeX- und LaTeX-Bestandteile und der entsprechenden Makro-Definitionen in LPLAIN.TEX, LATEX.TEX und LFONTS.TEX. Gerade das Wissen über letzteres File ist für die Einbindung eigener Zeichensätze wichtig. In dem Abschnitt über die LaTeX-Hauptstile finden sich die Angaben, die man braucht, wenn einem beispielsweise das normale Erscheinungsbild von Kapitelüberschriften nicht mehr genügt.

Es folgt ein Kapitel über die TeX-Programmierung, die man für LaTeX-Erweiterungen unabdinglich benötigt. Sie kann auf 80 Seiten sicherlich nicht die „Einführung in TeX" von Norbert Schwarz oder das Original-TeXbook von Donald E. Knuth ersetzen, ist aber dennoch als Referenz und erste Einführung geeignet.

Wie man eigene LaTeX-Layouts entwickelt, beschreibt Kopka anhand der refman-Option von Hubert Partl. Schrittweise wird gezeigt, wie eigene Layout-Vorstellungen in entsprechenden TeX-Code fließen und Seitenaufteilung, Überschriftengestaltung etc. gesteuert werden. Anhand von Briefgestaltungen, also eigenen letter.sty-Entwicklungen, zeigt Kopka, wie man die Optik von Briefköpfen umsetzt.

Der dritte große Teil des Buchs gibt eine Kurzeinführung in METAFONT. METAFONT ist der TeX-Bestandteil, mit dem die Zeichen sätze erzeugt werden. Dabei „programmiert" man die einzelnen Buchstaben - vereinfacht gesagt - durch die Festlegung von Stützpunkten und die Art der Linien. METAFONT ist nicht einfach und kaum jemand wird eine ernstzunehmende Schrift ohne tieferes typographisches Wissen erstellen können.

Dennoch sollte dieser Teil den Leser in die Lage versetzen, beispielsweise ein Firmenlogo oder ein exotisches Sonderzeichen selber zu erzeugen. Da immer mehr Zeichensätze als METAFONT-Code verfügbar sind, kann auch schon das Wissen um die Bedienung der Font-Erzeugung die typographischen Möglichkeiten mit TeX erweitern helfen.

Die Anhänge beschreiben das WEB-System, in dessen Pascal-Pseudocode TeX programmiert wurde und die Installation eines kompletten TeX-Pakets auf dem eigenen Rechner von der Erzeugung des eigentlichen Pascal-Codes mit WEB bis zum „Torture-Test", der die Korrektheit der Implementierung überprüft. Den Abschluß des Buchs bildet ein 17seitiges Register.

Das Buch deckt den ungeheuer weiten Bereich ab, der durch die Vielzahl von Hilfsprogrammen und Ergänzungen zu LaTeX entstanden ist. Es ist dadurch unbedingt zu empfehlen, will man ernsthaft mit LaTeX arbeiten und sich dabei über die vorgegebenen Möglichkeiten hinausentwickeln.

Über den seltsam nach unten verrutschten Seitenspiegel, die Verwendung der englischen, oberen Anführungszeichen und einige nicht aufgelöste Seiten- und Abschnittreferenzen im ersten Kapitel läßt sich problemlos hinwegsehen. Vielleicht wäre noch ein Abschnitt über die LaTeX-Output-Routine sinnvoll gewesen.

Mit diesem Buch ist die TeX-Reihe von Addison-Wesley fast komplett, wenn man auch merkt, daß sich der Verlag noch nicht zu einem deutschsprachigen METAFONT-Buch durchringen kann. LaTeX-Anwender sollten auf das Originalbuch von Leslie Lamport verzichten und sich die beiden Kopka-Bücher neben den Rechner stellen.

Inserentenverzeichnis

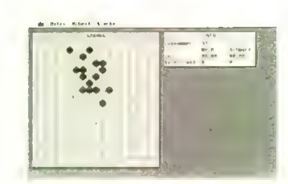
Immer up to date

	Programmname	Version	Daten	
	Adimens ST	3.1	N HM	
	Adiprog SPC Modula	1.1	N HM	
	Aditalk ST	3.0	N HM	
	Adress ST / Check ST	1.0	N H	
	Afusoft Morse-Tutor	2.0	N HML	
	Afusoft Radio-Writer	1.0	N HML	
	Afusoft Radiofax plus	1.0	N HML	1M
	AIDA	1.1	N HM	
	AnsiTerm	1.4	N	
	Assembler Tutorial	1.06	N HM	
	Banktransfer	1.0	N H	
	1st BASIC Tool	1.1	N HML	
	BTX/VTX-Manager	3.0	N H	1M
	Calamus	1.09	N H	1M
	Cashflow	1.0	N H	1M
	Chips At Work	1.0	N HM	
	CIS-L&G	2.1	N H	2M
	CiSystem	2.1		2M
	Creator	1.1	N H	
	Cubase	2.0		
	CW-Chart	8.0	N H	1M
	Daily Mail	1.2	N H	
	dBMAN	6.0	N HM	
☛	Diskus	2.0	N HM	
	dBMAN	5.10	N HML	
	Easybase	1.1	N HM	
	Easytizer	1.0	N HM	
	Easy Rider Assembler	2.04	N HM	
	Easy Rider Reassembler	2.31	N HM	
	Edison	1.00	N HM	
	fibuMAN	4.0	N H	
	fibuSTAT	2.3	N H	
	Flexdisk	1.4	N HML	
	FM-Meßtechnik	1.0.b	N HM	
	FTL Modula-2	1.18	N HM	
	Gadget	1.2.5b	N H	
	GEMinterface ST	1.1.	N HML	
	GFA-Artist	1.0	N L	
	GFA-Assembler	1.5	N HML	
	GFA-BASIC 68881	1.3	N HML	
	GFA-BASIC-Compiler	3.5	N HML	
	GFA-BASIC-Interpreter	3.5	N HML	
	GFA-Draft plus	3.01	N	
	GFA-Farb-Konverter	1.2	N H	
	GFA-Monochrom-Konverter	1.2	N ML	
	GFA-Objekt	1.2	N HM	
	GFA-Starter	2.0	N HML	
	GFA-Vektor	1.0	N	
	G+Plus	1.4	N HML	
	GrafStar	1.0	N H	
	Hänisch Modula-2	3.1	N HML	
	Hard Disk Accelerator	1.0	N HML	
	Hard Disk Sentry	1.10		
	Hard Disk Toolkit	2.0	N HM	
	Harddisk Utility	3.0	N HM	
	Harlekin	1.0	N H	1M
	Imagic	1.1	N HML	
	Intelligent Spooler	1.10	N HML	
	Interlink ST	1.89	N HM	
	ISI-Interpreter	1.20	N HM	
	Junior Prommer	2.33	N HM	
	K-Resource	2.0	N HM	
	Kleisterscheibe	2.2	N HM	
	Label ST	1.0	N HML	
	Laser C (Megamax)	2.1	N HML	
	1st_Lektor	1.2	N HM	
	Lern ST	1.22	N HML	
	Link_it GFA	1.1	N HML	
	Link_it Omikron	2.0	N HML	
	MagicBox ST	7.78	N HM	1M
	Mathlib	3.0	N HM	
	Mega Paint II	2.30	N H	1M
	Mega Paint II Professional	2.31	N H	1M
	Megamax Modula 2	3.5	N HM	
	MGE Grafikkarte	1.27	N	
	MGP GAL-Prommer	2.0	N H	
	Micro C-Shell	2.70	N HM	
	MPe II plus	1.02	N H	1M
	Mr Print	3.0	N H	
	MT C-Shell	1.2	N HM	1M
	Multidesk	1.82	N HML	
	Musix32	1.01	J H	
	NeoDesk	3.0	N HML	
	Notator	3.0		
	Omikron Assembler	1.86	N HML	
	Omikron BASIC-Compiler	3.06	N HML	
	Omikron BASIC 68881-Compiler	3.06	N HML	
	Omikron BASIC Interpreter	3.03	N HML	
	Omikron DRAW! 3.0	3.01	N HML	
	Omikron EasyGEM-Lib	1.0	N HML	
	Omikron Maskeneditor	1.0	N HML	
	Omikron Midi-Lib	2.1	N HML	
	Omikron Numerik-Lib	1.2	N HML	
	Omikron Statistik-Lib	1.5	N HML	
	PAM's TERM/4014	3.012e	N H	
	PAM's TurboDisk	1.7	N HML	
	PAM's NET	1.1	N HML	
	PCB-layout	1 19	N H	
	PC ditto Euroversion	3.96	N HML	
☛	PegaDress	1.0	N H	
☛	PegaFakt	2.0	N H	
☛	PegaStic	1.1	N H	
	phs-BTX-Box	6.1	N HML	1M
	phs-ST-Box	1.2	N HM	
	phs-Boxtalk	1.0	N HM	1M
	phs-Boxedi	1.0	N HML	1M
	Platon	1.45	N H	
	1st Proportional	3.13	N HM	
	Prospero Pascal	2.153	N HML	
	Prospero Fortran	2.153	N HML	
	Prospero C-Compiler	1.144	N HML	
	Prospero Developers Toolkit	1.111	N HML	
	Protos	1.1	N H	1M
	Querdruck2	2.05	N HM	
	Quick_Dialog	1 0	N HM	
	ReProk	1.10	N H	1M
	Revolver	1.1	N HML	1M
	Rufus	1.04	N HML	1M
	Scarabus	2.0	N H	
	Scigraph	2.0	J HM	
	Script	2.0	N HM	
	Search!	2.0	N HM	
	Signum! zwei	2.01	N H	
☛	Simula	2.1	N HML	1M
	Skylink	1.5	N H	1M
☛	Skyplot+	4.3	N H	1M
	Soundmachine II	1.0	N HM	
	SoundMerlin	1.01	N HM	
	SPC-Modula-2	2.0	N HML	
	Spectre 128	1.9	J HM	
	1st_Speeder 2	1.0	N HML	1M
	SPS ST	1.5	N H	1M
	STAD	1.3+	N H	
	Steuer-Tax 2.9	1.10	N HM	
	Steuer-Tax 3.9	1.10	N HM	
	STop	1.1	N HM	
	ST Pascal plus	2.08	N HM	
	Supercharger	1.4	J H	
☛	Technobox Drafter/2	2.0	J H	1M
☛	Technobox CAD/2-ST/TT	1.4	J H	2M
	Tempus Editor	2 10	N HM	
	That's Write	1.4	N HM	
	Theca Librarian	1.0	N HM	
	TIM	1.2	N H	
	TIM II	1.0	N H	1M
	Transfile ST 1600	1.1	N HM	
	Transfile ST 850	1.2	N HM	
	Transfile ST plus	3.1	N HM	
	Transfile ST E500	2.0	N HM	
	Transfile ST SF	2.0	N HM	
	Transfile ST IQ	1.4D	N HM	
	Turbo C	2.0	N HM	
	Turbo ST	1.8	N HML	
	UIS II + Hermes	2.5		
	V_Manager	3.1	N H	
	VSH Manager	1.0	N HML	1M
	WERCS Resource-Editor	1.0	N HM	
	Wordperfect	4.1	N H	
	Writer ST	1.4	N HM	
	Wordplus	3.15	N HML	

Irrtum vorbehalten! Daten-Legende : N = kein Kopierschutz, J = Kopierschutz, H = hohe Auflösung, M = mittlere Auflösung, L = niedrige Auflösung, 1M = mindestens 1 Megabyte, ☛ = Änderung gegenüber letzter Ausgabe



SONDERDISK



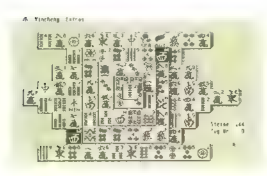


GOBANG

Ein Strategiespiel

GOBANG ist ein klassisches Brettspiel, bei dem abwechselnd Steine auf das Spielfeld gesetzt werden, wobei es gilt, 5 Steine in einer Reihe (senkrecht, waagrecht oder diagonal) zu plazieren. Der Computer bietet hier einen spielstarken Gegner, der nicht so leicht zu besiegen ist.
Neben dem Laden und Speichern einer Partie verfügt Gobang über verschiedene Spielstärken; vom Anfänger bis zum Profi. Auch die Blitzpartie, bei der jeder Spieler nur 30 Sekunden Bedenkzeit pro Spiel hat, bietet ihren speziellen Reiz. Ist man in einer schwierigen Lage, hilft der Rechner gerne mit einem Zugvorschlag aus.

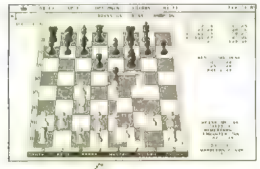
GOBANG
SD 49 DM 15



YINCHENG

Dieses Spiel beruht auf dem alten chinesischen Patience-Spiel Man-Jongg. Es geht darum, das mit 144 Spielsteinen gefüllte Spielfeld zu entleeren, wobei immer nur zwei zueinander passende und nach bestimmten Regeln positionierte Steine entfernt werden dürfen. YINCHENG beinhaltet eine zwei- und eine dreidimensionale Spielvariante, die sich zwar in den Regeln, doch kaum in der Spielqualität unterscheiden.

YINCHENG
SD 45 DM 20

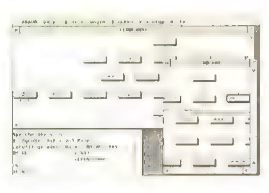


ODYSSEUS

Schachprogramm

Hinter Odysseus steckt ein spielstarkes und komfortables Programm. Die Züge lassen sich leicht per Maus eingeben. Es verfügt über eine Zeit- und eine Tiefensteuerung (bis zu 12 Halbzüge) und beherrscht den Turniermodus. Die beigefügte, jederzeit erweiterbare Bibliothek erlaubt dem Programm den Zugriff auf wichtige Züge. Mit ihm kann man Partien speichern, nachspielen und analysieren lassen.

Odysseus
SD 41 DM 25





BBAUM

BBAUM ist ein äußerst leistungsstarkes Tool für die Programmdokumentation von C-, PASCAL- und GFA-BASIC-Programmen. Vor allem die Einarbeitung in fremde Quelltexte wird vereinfacht, indem grafisch in Form eines Baumes die Funktions- bzw. Prozedurabhängigkeiten dargestellt werden.

BBAUM untersucht
- C-Quelltexte
- PASCAL-Quelltexte
- GFA-BASIC-Quelltexte (2.0, 3.0 und 3.5)
- DMP-Dateien (interne Baumstruktur)
- Verzeichnisse (Struktur Ihrer Festplatte/Diskette)

BBaum verwaltet Includes bzw. ausgelagerte Programmteile und fügt sie automatisch an die entsprechenden Stellen im Hauptprogramm an. Wahlweise werden auch die Routinen dargestellt, die in der System-Library definiert sind (z.B. *printf* oder *getchar*).
BBAUM ermöglicht weiterhin das Suchen eines bestimmten Namens und springt augenblicklich an diese Stelle in der Grafik. Gerade beim Erzeugen eines Directory-Baumes ist das sehr praktisch, da man auf diese Weise schnell eine bestimmte Datei findet.
BBAUM unterstützt alle 8-, 9- und 24-Nadeldrucker. Je nach Größe der Grafik werden auch mehrere aneinanderpassende Seiten bedruckt. Zur Druckzeitoptimierung ist zusätzlich eine direkte Ansteuerung der Centronics-Schnittstelle implementiert.
BBAUM unterstützt ebenfalls die Generierung von Funktionsprototypen, die den Umstieg auf den neuen ANSI-C-Standard erleichtern.

BBAUM
SD 50 DM 25

KOALA

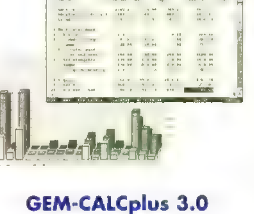
Der Monochromemulator

PANDA, der Farbemulator ermöglicht es, Software, die für Monochrommonitor geschrieben wurde (z.B. SIGNUM!), auch auf einem Farbbildschirm laufen zu lassen.

Freie Einstellung der Bildwiederholfrequenz. So kann man zwischen hoher Bildrate oder hoher Rechenleistung wählen • Bildaufbau während Diskettenzugriff abschaltbar • Bildschirm-Hardcopy auf Disk (Farb- und s/w-Bild)

KOALA ist kompakt, schnell und für alle ST-TOS-Versionen (1.0-1.6).

KOALA
SD 43 DM 15



GEM-CALCplus 3.0

Tabellenkalkulation

Überall dort, wo mit Zahlen hantiert wird, sei es zur betriebswirtschaftlichen Kostenrechnung, statistischen Auswertung von Meßreihen oder zur Erfassung der eigenen Finanzen, findet ein Kalkulationsprogramm seinen Einsatz. GEM-CALCplus ist ein flexibler und sehr leistungsfähiger Vertreter dieser Kategorie. Neben zahlreichen mathematischen und statistischen Funktionen bietet es eine exzellente Grafikausgabe der Daten als Kuchen-, Linien-, Balken-Stapel-, Säulen-, Block- und Flächengrafik.

Funktionen und Operatoren:
+, -, *, /, PI, DAT, ABS(), INT(), RND(), LOG(), EXP(), CLG(), SQR(), SIN(), COS(), TAN(), ASN(), ACS(), ATN(), FAK(), NUN(N:n), SUM(), AVE(), STA(), STD(), MUL(), MIN(), MAX(), QMW(), QMN()

GEM-CALCplus ist eine Weiterentwicklung des weitverbreiteten GEM-CALC (PD).

Die Erweiterungen:
• Arcussinus und Arcuscosinus • Blatt und Block schützbar • Fehlermeldungen mit Erläuterung • Suchfunktion • verbesserte Grafikdarstellung, Grafikausdruck und Grafik-Datenauswahl (Block) • flexible Speicherverwaltung • fixierbare Spalte • erhöhter Eingabekomfort • schnelleres Scrolling • u.v.a.m.

Alle Datenblätter können übernommen werden. (1MB sinnvoll)

GEM-CALCplus 3.0
SD 44 DM 25

DAME

Computerumsetzung des alten Brettspiels, wobei der ST einen spielstarken Gegner darstellt. Die Figuren werden per Maus angewählt, die Züge protokolliert und analysiert. Verschiedene Spielstärken, Zugvorschläge, Laden und Speichern einer Partie, sowie verschiedene Spielvarianten dürfen nicht fehlen.

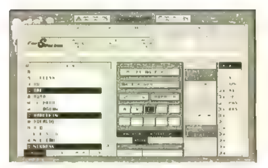
DAME
SD 29 DM 15

PANDA

Der Farbemulator

Der Farbemulator simuliert die Farbauflösungen des ST auf einem monochromen Monitor (SM 124,...). Dadurch kann man auch Farbprogramme laufen lassen, die sonst einen zweiten Monitor erfordern.

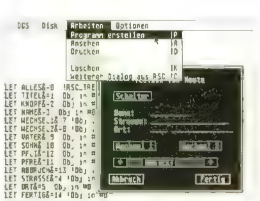
PANDA
SD 18 DM 15



FastSectorBackup 4.0

FastSectorBackup ist das ideale Tool für Ihre Datensicherung. Zum einen bietet es ein Image-Backup, welches komplette Partitionen sichert, und zum anderen ein sehr flexibles FileBackup. Damit lassen sich einzelne Dateien, welche nach Wildcards, Datum, Archiv-Bit oder einfach per Mausklick markiert werden, sichern. Weiterhin bietet FastSectorBackup die Möglichkeit, mehrere Backup-Vorgänge mit verschiedenen Markierungsarten in Batch-Dateien festzuhalten. Diese können dann automatisch ablaufen.

FastSectorBackup
SD 35 DM 25,-



Dialog Construction Set

für GFA-BASIC 3.x

Mit dem Dialog Construction Set (DCS) lassen sich auf einfache Art und Weise .ST-Dateien erstellen, die den Programmcode zur Behandlung von Dialogboxen unter GFA-BASIC 3.0 enthalten. So ist es möglich, diese schnell und bequem in eigene Programme einzubauen. Als Voraussetzung wird natürlich weiterhin das Resource Construction Set (wird bei GFA-BASIC mitgeliefert) benötigt. Einfach mit dem RCS erstellen und dann mittels DCS den Programmcode generieren. Grundkenntnisse über Dialogboxen und GFA-BASIC-Programmierung sind aber weiterhin erforderlich.

DCS
SD 48 DM 15

COMPLEX

Quiz

Quiz mit über 3500 Fragen aus den Wissensgebieten Geschichte, Geographie, Sport, Allgemeinbildung, Tierwelt, Kunst, Naturwissenschaft und Theater. Das Programm kann mit eigenen Fragen erweitert werden, somit steht die Möglichkeit zur Schaffung eines spezialisierten Quiz' (z.B. Motorwelt, Jura, Computerkunde oder gar Fremdsprache) offen. (1MB, 1-6 Spieler)

COMPLEX
SD 47 DM 20



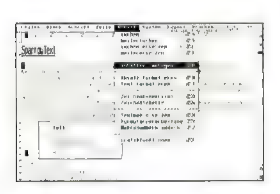
1stTrenn

vollautomatische Silbentrennung für 1stWordPlus

Darauf haben viele schon lange gewartet. Eine schnelle, automatische und präzise Silbentrennung für 1stWordPlus. 1stTrenn ersetzt die eingebaute Trennhilfe völlig, d.h. wird automatisch anstelle der eingebauten manuellen Trennung aktiviert (F10).

- arbeitet im Hintergrund (Accessory), 1stWordPlus muß nicht verlassen werden
- schnelle Trennung
- wahlweise mit Bestätigung oder vollautomatisch
- hohe Trefferquote von über 98%, d.h. etwa eine falsche Trennung bei 8 Seiten Text
- zusätzliche Autosave-Funktion des aktiven Textes
- läuft auf den deutschsprachigen 1stWordPlus Versionen 1.89, 2.02 und 3.15

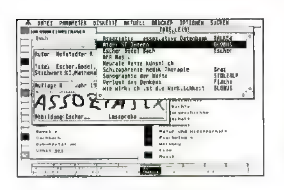
1stTrenn
SD 42 *DM 25.-*



SparrowText

Exklusives Textverarbeitungssystem mit besonderen Leistungsmerkmalen. Neben der Darstellung aller Schriftarten auf dem Bildschirm beherrscht es verschiedene Zeilenabstände, Proportionalschrift im Blocksatz (variables Spacing), verschiedene Font-Größen und vor allem einen eigenen Bildschirmzeichensatz. Damit lassen sich Sonderzeichen entwerfen und auch an den Drucker schicken.
SparrowText unterstützt das Zeichnen von Linien und Rechtecken, Trennung, Textformatierung, automatische Erzeugung eines Inhaltsverzeichnisses und ist vor allem sehr schnell dabei.
Als besonderen Leckerbissen ermöglicht es Formularverarbeitung, die sich hervorragend zum Ausfüllen von Briefbögen, Adreßfeldern oder allgemeinen Formularen eignet. Die Eingabefelder lassen nach Wunsch auch Eingabebeschränkungen (z.B. nur Zahlen) zu und bieten daher die Möglichkeit, gewisse Felder miteinander aufzuaddieren. Weiterhin kann man diese Felder automatisch ausfüllen lassen, da SparrowText Daten von einer Datenbank importieren kann und diese in die Felder einträgt. Dadurch läßt sich das Programm für Serienbriefe, Zeugnisse oder gar Rechnungen/Mahnungen einsetzen.

SparrowText
SD 37 *DM 25.-*



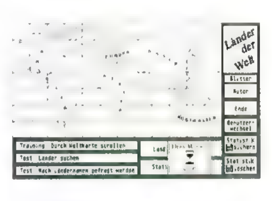
ASSOZIATIX

Assoziative Datenbank

Assoziatix ist eine assoziativ-Muster orientierte Datenverwaltung, die es ermöglicht, aus einer großen Datenmenge bestimmte Gruppen auszufiltern und daraus dank schneller assoziativer Suche nach bestimmten Konstellationen, Zusammenhänge zu finden (z.B. Rasterfandung).
Mit Hilfe des Formulareditors können alle Eingabemasken leicht am Bildschirm gestaltet werden, sogar mit Grafikeinbindung.

Einige Besonderheiten:
- Paßwortschutz, Export- und Importfunktion, Serienbriefe, Reportdokumentation, Statistische Berechnung numerischer Werte, Expertfunktion, Volltextsuche, Grafikeditor: Spiegeln, Drehen, Zoomen, Balken- Linien und Kuchengrafik

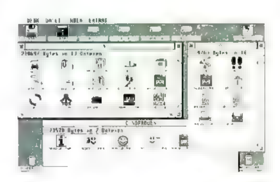
ASSOZIATIX (2 Disketten)
SD 27 a/b *DM 30.-*



LÄNDER DER WELT

Geographie-Lernprogramm mit leicht verständlicher Bedienung. 'Länder der Welt' vermittelt die Lage der einzelnen Länder auf der Weltkarte. Wo liegt z.B. Togo? 'Länder der Welt' hilft weiter und sorgt mit seinen Trainings- und Prüfungsfunktionen dafür, daß der Anwender diese Frage nicht ein zweites Mal stellen muß.

Länder der Welt
SD 39 *DM 15.-*



OPAQUE

Das Desktop mit neuem Gesicht

Wie wäre es mit einem zweckmäßigen und originellen Desktop? Opaque bietet die Möglichkeit, jedem Programm ein eigenes, sinnbezogenes Icon zuzuordnen. Auch die Laufwerke lassen sich ändern. Weiterhin kann man die Icons mit Wildcards definieren. Samt Icon-Editor und über 100 Icons.

OPAQUE
SD 22 *DM 15.-*



ICSIM

Logik-Simulator

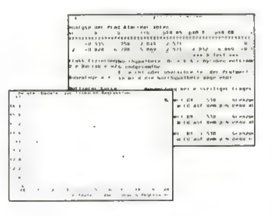
Das Programm simuliert das Verhalten von logischen Schaltungen. Bausteine und Verbindungen werden frei per Maus positioniert bzw. verbunden. Eine Schaltung läßt sich somit leicht austüfteln, testen und erst dann in die Praxis umsetzen.
Es sind die Logikbausteine nach DIN 40900 enthalten: AND, OR, NOT, NAND, NOR, XOR, RS-FF, KLEMME, LAMPE, SCHALTER, 0V und +5V.
Die Simulation wird als Impulsdiagramm oder Logiktabelle ausgegeben. Weiterhin liefert das Programm den Schaltplan und eine Liste der benötigten Bauteile.

ICSIM
SD 25 *DM 20.-*

DATEI LOGIK

Datenbank, die einfache Handhabung und große Flexibilität miteinander vereint. So ist es für jedermann möglich, sich ohne große Anstrengung eine Datenbank nach seinen Vorstellungen aufzubauen. Mit Hilfe des integrierten Formulareditors kann eine individuelle Abfragemaske erstellt, mit dem Etiketteneditor das Layout von Aufklebern oder Karteikarten für jeden Aufgabenbereich festgelegt und mit der Mailmerge-Funktion mit den Daten auch Serienbriefe erstellt werden.

Datei Logik
SD 36 *DM 20.-*

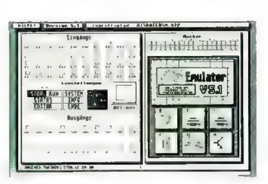


STatiST

modulares Statistik-Programmpaket

STatiST ist ein umfangreiches Paket zur Auswertung statistischer Daten. Zu jedem Prüfverfahren werden sämtliche Ergebnisse mit dem entsprechenden Wertungen und Kommentaren ausgegeben und, falls möglich, grafisch angezeigt. STatiST eignet sich für sämtliche, z.B. im Studium erforderlichen statistischen Auswertungen und macht das zeitaufwendige Rechnen per Hand und das Arbeiten mit Tabellen überflüssig.

STATIST (2 Disketten)
SD 32a/b *DM 30.-*



SPS-Emulator V 5.1

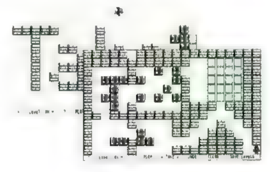
für programmierbare Steuerungen

Unser SPS-Emulator baut auf einem SIEMENS PG 605-Programmiergerät in STEP 5 auf. Mit ihm lassen sich SPS-Programme schreiben, auf Simulationsbasis austesten, laden, speichern, ändern, ausdrucken und als FUP (Funktionsplan mit logischen Gattern) ausgeben. Enthalten sind ein Editor, ein Interpreter und FUP-Generator.

Neu in der Version 5.1

Alle Befehle wurden voll im Siemens S5 Standart umgesetzt • 20 Timer als SE-, SA, SI-, SS-, SV-Timer zu verwenden • 20 Zähler (vorwärts/rückwärts), erhöhte Werte • Mehrfachzuweisungen nach einer Verknüpfung • wahlweise 20/40 Eingänge bzw. Merker • Schnellere Interpreterroutine (20-25%) • Startmerker für Autostart • Not-Aus-Merker/-Schalter • Blinkmerker. Vier Merker werden als astabiler Multivibrator angesteuert. • Sprungmarken (A-Z) • Komfortables Drucken der Awl • Die Merker-, Eingangs- und Ausgangsbezeichnungen können nun dezimal, hexadezimal oder byteweise bezeichnet werden • Klammerbefehl —> U(,) • Oder vor Und —> O • Neuer Texteditor • Erhöter Bedienungskomfort • Programmierung von Netzwerken

SPS Emulator V5.1
SD 14+ *DM 25.-*

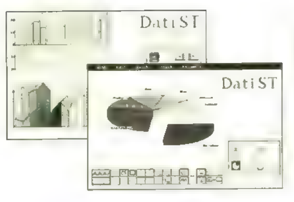


TAKE_1

Denkspiel

Die Regeln von Take_1 sind einfach - schieben Sie alle Diamanten auf die vorgesehenen Leerfelder. Leider gibt es ein paar Probleme... Der Vogel, den Sie mit den Cursortasten bewegen, kann immer nur einen Diamanten schieben und außerdem kann er sie nur schieben, nicht jedoch ziehen - das bedeutet, daß Diamanten an der Wand oder in einer Ecke nicht mehr verschoben werden können. Hier ist Taktik und Weitblick gefragt, denn so einfach, wie es im ersten Augenblick ausschaut ist es wirklich nicht. Durch den integrierten Level-Editor, lassen sich eigene Screens erzeugen, die dann, je nach Schwierigkeitsstufe und Anzahl der eingebauten Gemeinheiten, selbst ausgefuchste Profis schaudern lassen. Doch eignet sich das Spiel ebenso für die jüngere Generation, denn logisches Denken ist bekanntlich nicht nur Erwachsenen vorbehalten.

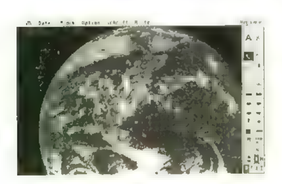
TAKE_1
SD 46 *DM 15.-*



DATIST
[illegible]asentationsgrafik

Grafiken sagen oft mehr als 1000 Zahlen, daher sollte man sich bei der Auswertung von Daten auf DatiST verlassen. DatiST stellt Ihre Daten als Kuchen-, Reihen-, Balken-, Säulen- und Liniengrafiken dar, entweder in 2D oder 3D, gefüllt oder als Rahmen. Lage, Größe, Dehnung und der Nullpunkt einer Grafik lassen sich frei mit der Maus einstellen, dafür sorgen die iconisierten Pop-Up-Menus. Im 3D-Modus kann gar die räumliche Perspektive frei variiert werden. Die so erzeugten Grafiken, lassen sich beschriften (z.B. mit S[illegible]GNUM!-Fonts) oder mit dem integrierten Zeichenprogramm bearbeiten, das vom Linienziehen über Blockoperationen bis hin zur Lupe alles bietet was man braucht. Um die Grafik zu Papier zu bringen bietet DatiST eine variable Druckeranpassung, die folgende Drucker unterstützt: Epson 9N/24N, NEC 24N, IBM PPR 24N, IBM AGM 24N, HP Laser, Atari-Laser!!

DatiST
SD 40 DM 25



Special Paint 2
[illegible]

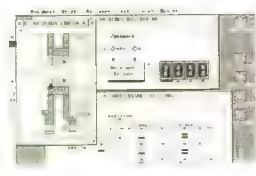
Grafikprogramm der Extraklasse. Neben den vielen nützlichen Funktionen zeichnet sich Special Paint vor allem durch seine Geschwindigkeit, seine bequeme Bedienung und seine Kompatibilität zu bekannten Malprogrammen aus. Special Paint bietet umfangreiche Blockfunktionen, Lasso, superschnelle Lupe, Maskierungen, Clippen, schnelle Bieg-, Zerr- und Drehoptionen, Animation und vieles mehr. Clipboardunterstützung, umfangreiche Textfunktionen (ladbare Fonts, Blocksatz, Zeilenumbruch).

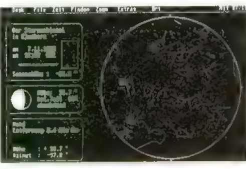
Special Paint
SD 21 DM [illegible]

ARIADNE

ARIADNE ist ein objektorientiertes Zeichenprogramm, d.h. Objekte können auch im Nachhinein ohne Auslösungsverlust verändert werden. Es bietet die Möglichkeit, jedes beliebige Grafikobjekt (mit Doppelklick) zu öffnen, worauf eine neue Zeichenebene bereitgestellt wird. Die Objekte auf dieser Ebene können dann wiederum geöffnet werden usw. Diese hierarchische Struktur eignet sich besonders zur Darstellung komplizierterer Dinge, z.B. Blockschaltbilder, Schaltungen etc.

ARIADNE
SD 8 [illegible]





ST-HIMMEL

Mit dem Programm kann der Anblick des Sternenhimmels für verschiedene Orte und Zeitpunkte berechnet werden. Ein ideales Programm für den Hobby-Astronomen.

Es zeigt:
- alle mit bloßem Auge (bei gutem Wetter) sichtbaren Sterne (~3000) mit Bezeichnungen, Helligkeiten und Entfernungen
- die mit bloßem Auge sichtbaren Planeten
- den Mond mit seiner Phase
- die hellsten Sternhaufen und Nebel
- einen Kometen
- die Höhe der Sonne über oder unter dem Horizont
- die Namen der sichtbaren Planeten
- die verschiedenen Sternbilder
- den Tierkreis
- die Eigennamen von 190 Sternen (z.B. Großer Bär statt Ursa Maior)
- die Tag- und die Nachtseite der Erde auf einer Weltkarte

ST-Himmel ist besonders anwenderfreundlich, so kann beispielsweise der Standort auf einer zoombaren Welt- bzw. BRD-DDR-Karte angeklickt werden.

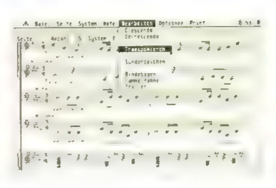
ST-HIMMEL
SD 38 DM 20

HARDCOPY II
[illegible] b Hardcopy für den ST

Universelles Hardcopy-Tool S/W- und **Farb**-Hardcopy auf *allen* Druckern in allen Größen, Screendump auf Disk, Formatkonvertierung, läuft als Acc[illegible], [illegible]bedienung, optimale D[illegible]

HARDCOPY II
SD 15 DM 15

1 nur für monochromen Monitor (SM 124)
2 nur für Farbmonitor



TRISTAN
Notensatzsystem

Für alle Musikfreunde, die nicht nur vom Blatt spielen, sondern auch aufs Blatt schreiben, bietet das Notensatzsystem TRISTAN die ideale Möglichkeit, ihre Noten professionell z[illegible] Papier zu bringen. Es lassen sich Partituren mit bis zu [illegible] mit max. [illegible] Notensysteme[illegible]. Alle im klassischen Notensatz gebräuchlichen Zeichen, u.a. auch Schla[illegible] und Vorschlagnoten lassen sich bequem mit der Maus edieren. [illegible]falls stehen mehrere Notenschlüssel, [illegible]mmelfahnen, Triller und Bindebögen zur Verfügung. Automatische Transponierfunktion, Ausdruck auf 9- und 24-Nadeldruckern in *maximaler* Druckerauflösung.

TRISTAN
SD [illegible]

FORMULA
[illegible]

Für mathematisch-wissenschaftliche Anwendung. Der eingebaute Formel-Interpreter beherrscht neben allen gängigen Operationen auch die Definition verschiedener Formeln in bestimmten Teilbereichen, logische Operationen und IF THEN ELSE. 3D-Grafiken lassen sich aus verschiedenen Blickrichtungen anzeigen und mit Schattierungen versehen.

FORMULA
SD 23 DM 20

Ultra-Disk
[illegible]

Ultradisk ist eine ultraschnelle, größenveränderbare resetfeste und reset-resistente RAM-Disk. Die Größe und die Laufwerkskennung kann frei bestimmt werden, und das alles ohne Inhaltsverlust und ohne den Rechner neu zu booten. Weiterhin enthalten ist ein ultraschnell[illegible], der dafür sorgt, daß [illegible] können [illegible] an den Drucker schickt. Auch darf [illegible] Maus[illegible] oder die Zeitanzeige [illegible] Bildschirmschoner nicht fehlen.

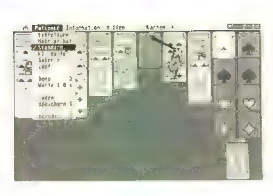
ULTRA-DISK
SD 33 DM 15

DER MOTOR

Der Motor erklärt mit zahlreichen Grafiken die Funktionsw[illegible] anschaulich sind die beweg[illegible] gezeigte Wissen wird [illegible] in einem Quiz abgefragt. Mit [illegible]geltem Katalysator!!

DER MOTOR
SD 20 DM 15

Sonderdisks unterliegen trotz des niedrigen Preises einem Copyright.



PATIENCE

Das Patiencespiel (patience = franz. Geduld) stammt aus Frankreich. Es ist ein Kartengeduldsspiel, das hohe Aufmerksamkeit erfordert. Es schult das Denkvermögen, fördert die Kombinationsfähigkeit, entspannt und beruhigt zugleich. [illegible] Programm sind folg[illegible] Varianten enthalten. [illegible] Zopf, Kleine Harfe, [illegible] Bildergalerie, Patience [illegible] nach bestimmten Regeln sortiert [illegible]zulegen. Sind alle Karten abg[illegible], gilt die Patience als gelöst. Das Programm gibt auf Wuns[illegible] lösungs[illegible] Eine ausführliche Anleitung z[illegible] Patiencen fehlt ebenfalls nicht.

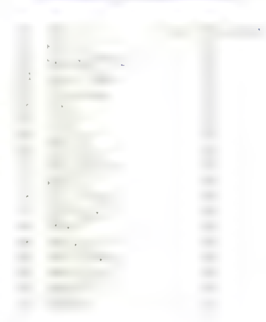
Patience
SD 11 DM 15

Programmierer aufgepaßt!!

Haben Sie nicht auch ein Programm geschrieben, das in diese Serie paßt? Sonderdisketten enthalten leistungsstarke Programme aus allen Bereichen zu günstigen Preisen. Als Autor erhalten Sie eine attraktive Umsatzbeteiligung. Lassen Sie doch mal was von sich hören.

MAXON Computer
Idee Sonderdisk
[illegible]straße 26
[illegible] Eschborn

Weitere Sonderdisks



Sonderdisk-Bestellung

Sonderdisks können Sie telefonisch oder schriftlich bestellen, oder nutzen Sie einfach die Bestellkarte im Heft.

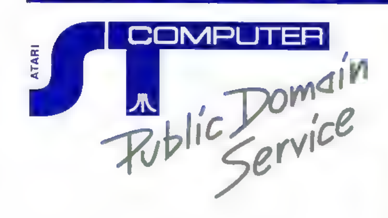


MAXON Computer
Schwalbacher Str. 52
6236 Eschborn
Tel. 06196/481811



Sonderdisks beinhalten Programme aus den verschiedensten Bereichen (z.B. Utilities, Grafik, Schulung, Spiele). Sonderdisks ermöglichen den Usern, qualitativ hochwertige Software zu einem kostengünstigen Preis zu erhalten. Im Preis ist eine Beteiligung der Autoren enthalten.

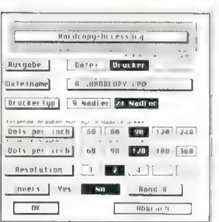
ST-COMPUTER PUBLIC DOMAIN



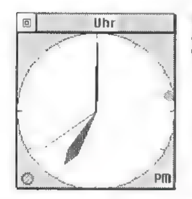
376 UTILITIES

TASCHEN: Dieses Programm stellt einen Taschenrechner dar. Alle wesentlichen mathematischen Funktionen sind enthalten. Es können mit dem Taschenrechner auch Vergleichsrechnungen durchgeführt werden (ACC/PRG).

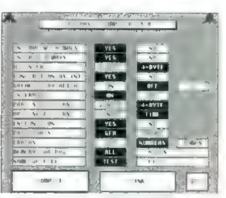
FORMAT: Formatierprogramm bis 11 Sektoren 84 Tracks (ACC/PRG).



HARDCOPY: Hardcopy-Programm, mit verschiedenen Auflösungsstufen für EPSON-kompatible Drucker (ACC/PRG).



UHR.ACC: Analog-Uhr (sekundengenau) mit Wecker (ACC/PRG).



COMPILER: luxuriöse Shell für GFA-Interpreter/ Compiler/Linker (s/w)

EDITOR.ACC: einfacher Editor als ACC

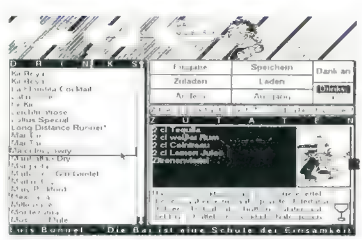
BBSP: Multitool mit folgenden Fähigkeiten: Stellen der Systemuhrzeit, schnelle RAM-Disk, schnelle Hardcopy (auch Ausschnitt), Disk-Verify an/aus, Zeit und Datumsanzeige, Bildschirmschoner, progressive Quick-Maus, Alarm. (s/w)

COMPARE: einfaches Programm zum Vergleichen zweier Dateien

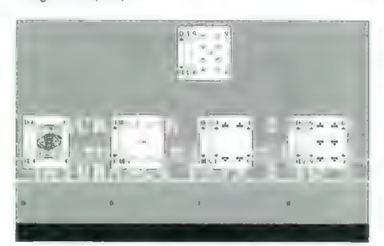
FILER: dient zum Anschauen und Ändern beliebiger Dateien. Darstellung als ASCII und im Hexcode. Blockoperationen.

SCRAMBEL: Dateiverschlüsselung

377 SPIELE, MUSIK UND GETRÄNKE



BAR: neueste Version der bekannten Drink-Datenbank. Unentbehrlich für jede Party oder gar zur Jahreswende. Enthält über 150 verschiedene Drinks. Die Zutaten können frei eingegeben werden, die Wirkung des Drinks wird durch ein entsprechendes Bild dargestellt. Der Barkeeper unterhält Sie durch kräftiges Schütteln, Mixen und entsprechende Kommentare während der Arbeit (bzw. des Vergnügens) mit dem Programm. (s/w)

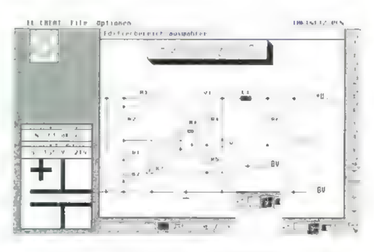


BLACK-JACK: Dieses Programm ist eine Simulation des Spieles „Black Jack" (die etwas feinere Version von 17 & 4). Die Regeln, nach denen gespielt wird, entsprechen denen in Spielcasinos, dabei übernimmt der Computer die Rolle des Croupiers. Bis zu 4 Spieler, verschiedene Einsätze und diverse Regelbestimmungen (Splitten, Anz. Decks, Versichern,..) machen das Programm zur echten Casino-Simulation. (s/w)

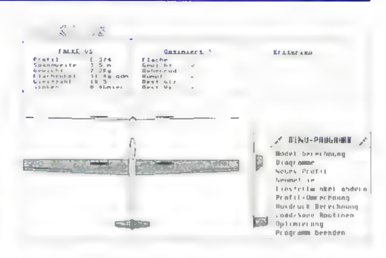
OHR_HELDEN: Programm zur Gehörschulung. Über Lautsprecher oder über MIDI werden Töne ausgegeben, die man bestimmen muß. (s/w)

MATCH_IT: TETRIS-Variante mit variablem Urzustand, d.h., daß das Feld bereits mit bestimmten Figuren vorbelegt ist. Spiel per Joystick. (s/w)

380 TECHNIK

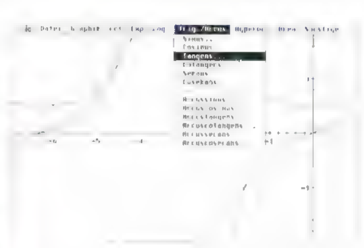


EC_CREAT: komfortabler Schaltplanzeichner (Electronic Circuit Creator) mit umfangreicher Bauteilbibliothek. Sämtliche grafischen Grundfunktionen (Linien, Box, Lupe,..) sind vorhanden. Ein Bauteil-Editor (128*128 Pixel) sorgt für beliebige Erweiterbarkeit der Bibliothek. Unterstützt 8- und 24- Nadeldrucker. (s/w)



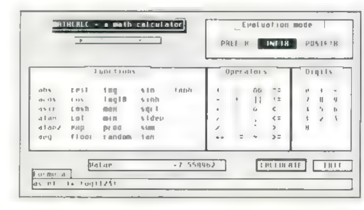
OPT_XX: Programm zur Berechnung von Segelflugzeugdaten. Diagramme über Gleitzahl, Sinken, Widerstandspolare und Stabilitätsdiagramm geben Aufschluß über die Konstruktion. Geometriedarstellung einer Dreiseitenansicht vom Modell, Profilumrechnung, Optimierung. (s/w)

381 MATHEMATIK



GRUNDPLOT: Plot-Programm für mathematische Grundfunktionen (Potenz-, Exponential- und Logarithmus-, trigonometrische und Arcus-, hyperbolische und Area-, Betrag-, Signum- und Entierfunktion), ihrer 1. und 2. Ableitung sowie eine Integralfunktion. Dies kann sowohl für die gewöhnlichen [z.B. f(x)=sin(x)] als auch für die allgemeinen [z.B. f(x)=a*sin(bx+c)+d] Funktionen geschehen. (s/w)

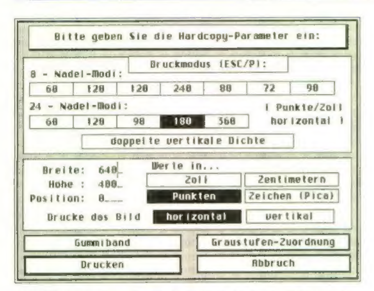
PLANIMET: Flächenberechnungsprogramm. Berechnet den Inhalt geschlossener Flächen, die entweder selbst gezeichnet oder als fertiges Bild geladen werden. Ähnlich der Füllroutine wird die Fläche ausgefüllt, doch dabei berechnet. Skalierungsfaktor und Maßstab erlauben z.B. die Berechnung gescannter Vorlagen.



MATHCALC: erlaubt die Definition von Variablen und Funktionen. So definiert die Angabe von test (a,b):=a+10*b eine neue Funktion. Boolsche Funktionen, Bit-Shifts, Vergleiche und natürlich die gängigen mathematischen Operatoren machen das Programm sehr wertvoll. Als PRG oder TTP nutzbar.

382 DRUCKER-UTILITIES

SPOOLER: Drucker-Spooler als ACC. Spoolt Text- und Hardcopy.



HARDCOP: flexible HARDCOPY. Gummiband, Anpassung an 8- und 24-Nadeldrucker in verschiedenen Dichten, Graustufenkonvertierung machen dieses Programm sehr interessant.

HARDSPEZ: System-Hardcopy für 8 Nadeldrucker. Ausschnittsbestimmung, Orientierung, Vergrößerung.

HPIII: Installationsprogramm für HPLASER III. Besonders interessant ist die Einstellung der skalierbaren Schriften dieses Druckers.

HPLIST: Druckprogramm für ASCII-Dateien unter Berücksichtigung der Sonderzeichen, Druckrand und Font.

PFFR: Drucker-Spooler, der durch sein interruptgesteuertes Prinzip extrem schnell ist bzw. das System wenig aufhält.

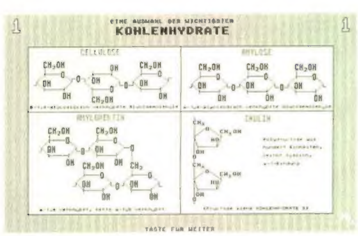
SCHWUPP: Konverter von Protext nach SIGNUM!. Übernahme der Attribute, Zeichenkonvertierung und Absatzerkennung.

BENUTZERLEX: Stark erweitertes Rechtschreiblexikon für Beckertext mit ca. 4500 Wörtern.

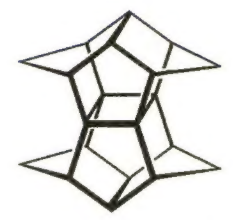
STARLC24: Programm zur Einstellung dieses Druckers (s/w)

POSTERTEXT: Programm zum Drucken von Miniaturbildern, Postern und Spruchbändern. Für 8- und 24-Nadeldrucker. (s/w)

PD 383 CHEMIE



LELEB: umfangreiche Berechnungen aus der Chemie. Vom Molgewicht über ph-Wert, Konzentrationen und Gleichgewichtsberechnungen bis hin zu thermischen Zustands- und van der Waals-Gleichungen. Weiterhin sind umfangreiche Informationstafeln eingebunden. (s/w)

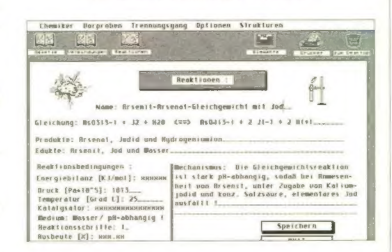


OC_GRAF: Malprogramm mit speziellen Funktionen zum Zeichnen von chemischen Elementen. So lassen sich einfachst diverse Ringe in verschiedenen Größen (z.B. Sechsring in Sesselform) zeichnen und stufenlos drehen. Gut geeignet zum Einbinden in Textprogramme zur Unterstützung chemischer Referate. (s/w)

PSE: Periodensystem der Elemente. Weitere Fragen? (s/w)

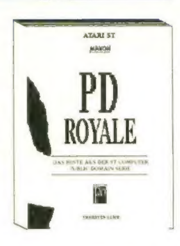
PD 384 CHEMIKER 2.2

CHEMISCHE DATENSAMMLUNG



CHEMIKER ist ein nützliches Programm, das Aufschluß über chemische Elemente, Gesetze, Reaktionen, Verbindungen, Analyse- und Verarbeitungsverfahren gibt. Darunter fallen z.B. Soda-Auszug, Oxidationsschmelze, Boraxperle, Flammenfärbung, Aufschlüsse, Anionen\Kationen oder diverse Trennungsgänge aus Salzsäure-, Kupfer-, Arsen-Zinn-, Schwefelwasserstoff-, Ammoniumsulfid-, Ammoniumcarbonat-Gruppe oder Lösliche Gruppe.
Nicht zuletzt die Periodentabelle mit chemischen und allgemeinen Daten zu allen Elementen. (s/w)

Heiligabend naht!



"PD ROYALE - Das Beste aus der ST Computer Public Domain Serie" entweder direkt bei MAXON Computer für DM 34,- inkl. Versandkosten oder über den Buchhandel (unverbindlicher Verkaufspreis DM 29,-, ISBN 3-927065-07-2).

Die in PD ROYALE beschriebenen Programme sind auf über 60 Disketten verteilt. Wir haben uns daher dazu entschlossen, diese Programme neu gesammelt zusammenzustellen. Herausgekommen sind drei *'Pakete Royale'*, bestehend aus je 5 randvollen Disketten, gefüllt mit Programmen aus diesem Buch. Damit Sie je nach Ihren Interessensgebieten auswählen können, haben wir themenorientierte Pakete zusammengestellt.



PAKET ROYALE A - Anwendungen
Minitext, Deluxe Fontmaster, Fontedit, Tiny Editor, Keyhelp, Brief, Last Word, Umlaut, Prn-Send, Speedwriter, ASCII-Edit, Printing Press, Little Painter, Showtime, Sticker, LQ 800, Hardcopy, Public Painter, WDR-Bild, SBASE, GEM-Calc.

5 Disketten DM 39,-



PAKET ROYALE B - Utilities+Unterhaltung
ARC, Sagrotan, Bitte ein Bit, Cruncher, FCOPY III, Hyperformat, Packer, ZOO, Calendar, Chooseboot, Deskedit, Dump, FSelect, Give Up, Goodies, James, JClock, Lock, Megamatic, Simple, ST-Klick, Termin, Uhr, Werkzeugkiste, X-Utility, CHR-Tree, Diskkatalog, Goodview, Hide, SuperFile-Copy, XDir, XDirList, Bar, Biorythmus, ST-Kalender, Kalender, Shapes, Erdkugel.

5 Disketten DM 39,-



PAKET ROYALE C - Spiele
Tetrix, Stones, Columns, Diamond Miner, Solitaire, U-Boot, Laserschach, Superbreakout, Invaders, Mac Pan, Memory, Quiz, Wizzy, Go-Up, DGDB I+II

5 Disketten DM 39,-

ABKÜRZUNGEN

1MB = mind. 1MB Speicher notwendig
s/w = nur Monochrom; f = nur Farbe



DIREKT-VERSAND

Alle PD-Disketten unserer Sammlung gibt es nur direkt bei MAXON-Computer.

1. Schriftliche Bestellung

- Der Unkostenbeitrag für eine Diskette beträgt DM 10,-
- Hinzu kommen Versandkosten von DM 5,- (Ausland DM 10.-)
- Bezahlung per Scheck oder Nachnahme
- (Im Ausland nur Vorauskasse möglich)
- Bei Nachnahme zuzüglich DM 4,00 Nachnahmegebühr
- Ab 5 Disketten entfallen die Versandkosten (DM 5.- bzw. DM 10.-)
- Der Versand kann aus technischen Gründen **ausschließlich** gegen Nachnahme oder Vorauskasse erfolgen (auch für Händler!).

2. Telefonische Bestellung

MAXON-Computer GmbH
'PD-Versand'
Tel.: 0 61 96 / 48 18 11
Fax: 0 61 96 / 4 18 85
Mo-Fr 9^00 - 13^00 und 14^00 -17^00 Uhr

- Lieferung erfolgt per Nachnahme

Adresse:

MAXON-Computer GmbH
'PD ST-Computer'
Schwalbacher Strasse 52
D-6236 Eschborn

Nutzen Sie die PD-Karte in diesem Heft

In der nächsten ST-Computer lesen Sie unter anderem

Steve - nicht nur ein Vorname

Seit 1986 gibt es für den Atari ST ein Programmpaket, das völlig zu unrecht etwas ins Abseits geraten ist: STEVE, zusammengestellt unter Federführung von Primoz Jakopin, wohnhaft in der Gegend von Ljubljana, Jugoslawien, vertrieben von der Firma Kieckbusch. Wir zeigen Ihnen, was dieses interessante Programm leistet. In Ausgabe 2 der ST-Computer.

Tempus Word - totgeglaubt

Tempus Word lebt - wie Sie in der nächsten Ausgabe in unserem ultimativen Testbericht lesen können. CCD hat endlich die endgültige Version auf den Markt gebracht. Ob das Programm schon aus den Kinderschuhen ist und was es leistet, verrät Ihnen die Februar-Ausgabe der ST-Computer.

Buttondesign - Mac läßt grüßen

Wenn Sie schon einmal versucht haben, eigene Buttons zu gestalten und dann gar noch in eigenen Programmen zu benutzen, werden Sie an den Rand der Verzweiflung getrieben worden sein. Aber wieso soll man es schwer machen, wenn es auch einfacher geht? Mit unserem Programm „Buttondesign" sind selbstgestaltete Knöpfe kein Problem mehr! Und nach dem Entwurf können Sie die Buttons gleich in C einbinden. Ein Grund mehr, sich die nächste ST-Computer schnellstens zu besorgen!

Courier HST Dual Standard

Was, Sie haben noch ein Modem mit 2400 Baud? Dann sind Sie nicht mehr auf dem neuesten Stand der Dinge, denn es gibt Modem-Hersteller, die Geräte mit 9600 oder gar 14400 Baud anbieten. Das Wunschgerät eines jeden DFÜ-Freaks, das Courier HST Dual Standard, stand einen Monat lang bei uns auf dem Prüftisch. Lesen Sie in der Februar-Ausgabe, warum Sie sich nicht mehr von diesem Gerät trennen können!

Die nächste ST Computer erscheint am Fr., den 25.01.91

Fragen an die Redaktion

Ein Magazin wie die ST-Computer zu erstellen, kostet sehr viel Zeit und Mühe. Da wir weiterhin vorhaben, die Qualität zu steigern, haben wir Redakteure eine große Bitte an Sie, liebe Leserinnen und Leser: Bitte haben Sie Verständnis dafür, daß Fragen an die Redaktion nur **donnerstags von 14[00]-17[00] Uhr** unter der Rufnummer 06196/481814 telefonisch beantwortet werden können.

Natürlich können wir Ihnen **keine** speziellen Einkaufstips geben. Wenden Sie sich in diesem Fall bitte an einen Fachhändler. Wir können nur Fragen zur ST-Computer beantworten.

Vielen Dank für Ihr Verständnis!

Impressum ST Computer

Chefredakteur: Harald Egel (HE)

Redaktion:
Harald Egel (HE)
Joachim Merz (JM)
Dieter Kühner (DK)
Martin Pittelkow (MP)

Redaktionelle Mitarbeiter:
C.Borgmeier (CBO)
Claus Brod (CB)
Ingo Brümmer (IB)
Derek dela Fuente (ddF)
Stefan Höhn (SH)
Raymund Hofmann (RH)
Claus P. Lippert (CPL)
Chr. Schormann (CS)
U.Seimet (US)
R.Tolksdorf (RT)
Thomas Werner (TW)

Autoren dieser Ausgabe:
M. Baldauf
L. Bauer
D. Brockhaus
M. Chakravarty
C. Conrad
T. Doll
S. Dreckmann
G. Ekart
C. Fihl
P. Köster
R. Kotzian
S. Krüppel
D. Rabich
O. Völckers

Auslandskorrespondenz:
C.P.Lippert (Leitung), D.Dela Fuente (UK)

Redaktion: MAXON Computer GmbH
Postfach 59 69
Industriestr. 26
6236 Eschborn
Tel.: 0 61 96/48 18 14, FAX : 0 61 96/4 11 37

Verlag: Heim Fachverlag
Heidelberger Landstr. 194
6100 Darmstadt 13
Tel.: 0 61 51/5 60 57, FAX : 0 61 51/59 10 47 + 5 60 59

Verlagsleitung:
H.J.Heim

Anzeigenverkaufsleitung:
U.Heim

Anzeigenverkauf:
K.Margaritis

Anzeigenpreise:
nach Preisliste Nr.5, gültig ab 1.3.90
ISSN 0932-0385

Layout:
Manfred Zimmermann, Angela Heischmann

Titelgestaltung:
Axel Weigend

Fotografie:
Martin Pittelkow, Klaus Ohlenschläger

Illustration:
Manfred Zimmermann

Produktion:
B.Kissner

Druck:
Frotscher Druck GmbH

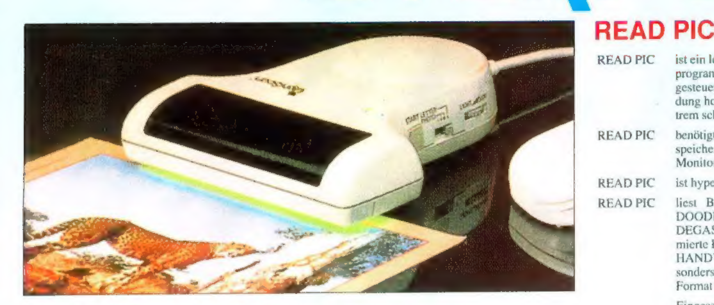
Lektorat:
V.Pfeiffer

Bezugsmöglichkeiten:
ATARI-Fachhandel, Zeitschriftenhandel, Kauf- und Warenhäuser oder direkt beim Verlag

ST Computer erscheint 11 x im Jahr
Einzelpreis: DM 8,-, ÖS 64,-, SFr 8,-
Jahresabonnement: DM 80,-
Europ. Ausland : DM 100,- Luftpost : DM 130,-
In den Preisen sind die gesetzliche MWSt. und die Zustellgebühren enthalten.

Manuskripteinsendungen:
Programmlistings, Bauanleitungen und Manuskripte werden von der Redaktion gerne angenommen. Sie müssen frei von Rechten Dritter sein. Mit seiner Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung auf Datenträgern der MAXON Computer GmbH. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.

ATARI ST



GENISCAN GS4500 ST

- ❑ Der einfach einzusetzende Handy-Scanner mit 105 mm Scanbreite und 400 dpi Auflösung ermöglicht die Reproduktion von Grafik und Text auf dem Schirm.
- ❑ Ein leistungsfähiger Partner für Desktop-Publishing-Anwendungen.
- ❑ Zum Lieferumfang gehört der GS4000Scanner sowie die Schnittstellen- und Editiersoftware.
- ❑ Mit Geniscan können Sie auf einfache Weise Bilder, Texte und Grafiken in den ST einlesen.
- ❑ Helligkeit und Kontrast einstellbar.
- ❑ Die leistungsfähige Software erlaubt Kopieren und Einfügen von Darstellungen.
- ❑ Speichert Darstellungen in Formaten ab, die sich für DEGAS, NEOCHROME, FLEETSTREET und andere eignen.
- ❑ Ausdrucke mit allen Epson-Kompatiblen möglich.
- ❑ Unerreichte Möglichkeiten beim Einlesen und Editieren zu einem unschlagbaren Preis.

Jetzt inkl. Zeichenprogramm THE ADVANCE OCP ART STUDIO.

einschließlich Soft- und Hardware.
Zusätzliches Interface
Software für PC DM 99,- zzgl. DM 10,- Versandkosten



nur **DM 498,-**
zzgl. DM 10,- Versandkosten

READ PIC

READ PIC ist ein lernfähiges Texterkennungsprogramm, es ist vollständig GEM-gesteuert und durch die Verwendung hochoptimierter Routinen extrem schnell in der Texterkennung.

READ PIC benötigt mindestens 400 KB Arbeitsspeicher und einen monochromen Monitor.

READ PIC ist hyperscreen-fähig.

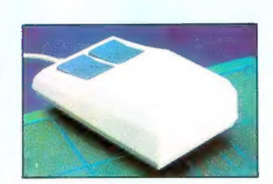
READ PIC liest Bildschirmformat-Bilder im DOODLE und im PI 3-Format von DEGAS. Es kann aber auch komprimierte Bilder im STAD-Format, im HANDY-Painter-Format, aber besonders im Standard-GEM-IMG-Format übernehmen.

Eingescannte Bilder können unkomprimiert als DEGAS-PI 3-Bild oder in voller Größe im GEM-IMG-Format abgespeichert werden. Vom eingescannten Bild kann darüber hinaus eine Hardcopy erzeugt werden (nicht im hyperscreen-Modus).

READ PIC kann überlappende Buchstaben (bis zu drei) trennen und ist auch in der Lage, verschmolzene Buchstaben bzw. echte Ligaturen zu verarbeiten. Die erkannte Schrift kann als Textdatei auf Diskette abgespeichert werden. Bei genügend Speicherplatz kann die erkannte Schrift direkt mit einem Texteditor Ihrer Wahl nachbearbeitet werden.



nur **DM 150,-**
zzgl. DM 10,- Versandkosten

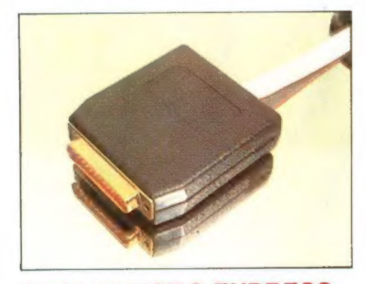


GENIUS-MAUS: Die Maus-Alternative

- ❑ Voll Amiga-kompatibel
- ❑ Gummibeschichtete Kugel
- ❑ Optische Maus
- ❑ Semi-optische Maus
- ❑ Inklusive Maus-Matte

Komplettpaket

nur **DM 79,50**
zzgl. DM 10,- Versandkosten



NEU SYNCRO EXPRESS

SYNCRO EXPRESS ist der Nachfolger von unserem bekannten A-COPY ST. Es ist eine Neuentwicklung auf dem Gebiet des Kopierverfahrens. SYNCRO EXPRESS macht eine Sicherheitskopie von fast allen Originalen. SYNCRO EXPRESS kopiert eine ganze doppelseitige Diskette in 40 Sekunden. SYNCRO EXPRESS funktioniert nur mit einem zweiten Laufwerk. SYNCRO EXPRESS ist ein steckbarer Hardwarezusatz mit der dazugehörenden Software für die Angabe der Start- und Endtracks sowie der Seitenwahl.

Preis **DM 99,-**
zzgl. DM 10,- Versandkosten

Als Update für A-COPY ST Preis: **DM 79,-**
zzgl. DM 10,- Versandkosten

A-COPY ST

Kopierprogramm.
Vollständiges Kopieren von Disks. Selbst aufwendig geschützte Programme werden in unter 60 Sekunden kopiert.

Preis **DM 69,-**
zzgl. DM 10,- Versandkosten



ST SUPER TOOLKIT II™

Ein Paket leistungsfähiger Dienstprogramme für alle ST-Modelle.

- ❑ Track- und Sektoreditierung mit bis zu 85 Tracks und 255 Sektoren.
- ❑ Eine Such- und Ersetzfunktion ersetzt automatisch einen angegebenen Wert mit einem neuen.
- ❑ Ein Werkzeug, das die hohe Auflösung nutzt. Arbeitet nur mit dem monochromen Monitor in der höchsten Auflösungsstufe.
- ❑ Im Info-Modus werden alle wichtigen Daten angezeigt.
- ❑ Fünf unterschiedliche Editorbetriebsarten – Laufwerks-, Disk- oder Datei-orientiert. Direkte Anwahl von Boot- und Directorysektoren möglich.
- ❑ Vollständig menü-/piktogrammbedient. Die Disk kann direkt im Hex- oder ASCII-Format editiert werden.
- ❑ Vergleichsfunktion – vergleicht zwei Disketten und zeigt die Unterschiede an. Das richtige Werkzeug für den Disk-Hacker.
- ❑ Umfangreiche Druckerunterstützung mit Hilfe einer Parameterbox.

nur **DM 49,-**
zzgl. DM 10,- Versandkosten



ATARI ST-LAUFWERKE

- ❑ Komplett anschlußfertig.
- ❑ Voll abgeschirmt durch Metallgehäuse.
- ❑ Atarifarbene Frontblende und Lackierung.
- ❑ Abschaltbar.
- ❑ 3 ms Steprate.
- ❑ 5,25"-Drives umschaltbar 40/80 Tracks.
- ❑ Kapazität 720 KB, 2 x 80 Spuren.
- ❑ Mit Bedienungsanleitung und 6 Monate Garantie.
- ❑ mit Track-Display

Preis: 5,25"-Drives ohne Track-Display **DM 229,-**
3,5"-Drive mit Track-Display **DM 199,-**
3,5"-Drive ohne Track-Display **DM 179,-**
zzgl. DM 10,- Versandkosten



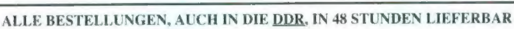


NEU! VOLLOPTISCHE MAUS

- ❑ Volloptische Maus.
- ❑ Sehr hohe Auflösung (250 dpi), für sehr genaues Arbeiten.
- ❑ Keine mechanische Teile (kein Verschleiß und Verschmutzung).
- ❑ Direkt anschließbar.
- ❑ 100% kompatibel.
- ❑ Inklusive Maus-Matte.

Preis: nur **DM 119,-**
zzgl. DM 10,- Versandkosten

ALLE BESTELLUNGEN, AUCH IN DIE DDR, IN 48 STUNDEN LIEFERBAR



EUROSYSTEMS

Hühnerstr. 11, 4240 Emmerich, Tel.: 028 22/45589 u. 45923
Telefax 0031/8380/32146, Tag- & Nacht-Bestellservice
Auslandsbestellungen nur gegen Vorauskasse

BESTELLUNG BEI VORKASSE DM 6,-, NACHNAHME DM 10,-
Versandkosten, unabhängig von der bestellten Stückzahl.

Distributor für Berlin: Mükra Datentechnik, Schöneberger Str. 5, 1000 Berlin 42, Tel.: 030/7529150/60
für Österreich: Computing Zechbauer, Schulgasse 63, 1180 Wien, Tel.: 0222/4085256
Rechner-Ring, Grazer Str. 90, 8605 Karpfenberg, Tel.: 03862/24950
für die Schweiz: Swiss Soft AG, Obergasse 23, CH-2502 Biel, Tel.: 032/231833
für Holland: Eurosystems NL, Postbus 179, 6710 BD Ede, Tel. 085/516565

Mit Erscheinen dieses Heftes verlieren ältere Preise ihre Gültigkeit.

EIN GUTER FREUND





MORTIMER PLUS

Für viele unserer Kunden ist Mortimer ein guter Freund geworden. Er war stets da, wenn er gebraucht wurde; verstand sich gut mit allen anderen Programmen – und packte immer kräftig mit an. In diesem Jahr hat er nochmals kräftig dazugelernt. Und ist so – wie wir meinen – ein noch besserer Freund geworden. Näheres erfahren Sie im Prospekt oder telefonisch.

Mortimer Plus DM 129,– Upgrade DM 60,–

Mortimer DM 79,– (Upgrade bis zum 31. 12. 90 DM 50,–)

NEUHEITEN

+ Texteditor mit automatischem Zeilenumbruch, Blocksatz und Menüzeile
+ Speichermonitor: Daten retten nach Absturz beliebiger Programme
+ Dateiauswahlbox ins Betriebssystem eingebunden
+ erweiterter Tastaturmakro-Treiber
+ lauffähig auf ATARI TT
+ Uhrzeit einstellen & über Kaltstart retten
+ trotzdem weniger als 80 Kbyte – kein Problem selbst für einen 520 ST

Mortimer Plus kann natürlich alles, was Mortimer kann – und das ist eine ganze Menge.

OMIKRON.Soft + Hardware
Sponheimstr. 12a · D-7530 Pforzheim
Telefon 07231/356033



XEST, Webgasse 21, A-1060 Wien
OMIKRON. France, 11, rue dérodé, F-51100 Reims
Elecomp, 11, avenue de la gare, L-4131 Esch/Alzette
Jotka Computing, Postbus 8183, NL-6710 AD Ede